1

A USER DEFINABLE SLAM AIRFIELD MODEL
DESIGNED FOR
EXPERIMENTATION AND ANALYSIS
VOLUME I

THESIS

AFIT/GST/OS/82M-9 Robert W. Mann
Maj USAF
Brian J. Shook
2Lt USAF

Approved for public release; distribution unlimited.

AFIT/GST/OS/82M-9

Thesis

A USER DEFINABLE SLAM AIRFIELD MODEL
DESIGNED FOR
EXPERIMENTATION AND ANALYSIS
Volume I

by

Robert W. Mann
Maj USAF

and

Brian J. Shook
2Lt USAF

Prepared in partial fulfillment of requirements for a Master's Degree

March 1982

School of Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base
Ohio

Approved for public release; distribution unlimited.

PREFACE

The question of how best to target airfields has long been a subject of debate. To this date no satisfactory solution has been found. This thesis presents a new approach to the targeting problem which is conceptually more straightforward than other approaches which have used computer simulation. This approach, properly applied, offers to yield a solution to the targeting problem which is both defendable and repeatable.

We wish to gratefully acknowledge the assistance of Major Rick Schmitt, Tactical Air Warfare Center, Future Plans Dvision, in providing background information and advice on thesis modeling efforts. We sincerely appreciated the assistance of Major Jack Bogusch, AFIT, GST-83M, in critiquing all the input data for the normal scenario from the perspective of his extensive fighter background.

We also wish to thank Lieutenant Colonel Saul Young, our faculty and thesis advisor; and Lieutenant Colonel Tom Clark, our reader and computer simulation instructor for two quarters, for their invaluable constructive criticism and support throughout this research effort.

Finally, we wish to thank Nancy, Allison, and Amy Mann for their patience and understanding, which were greatly appreciated.

Robert W. Mann
Brian J. Shook

TABLE OF CONTENTS

Volume I

Page

Volume II

Volume III

LIST OF FIGURES

AFIT/GST/OS/82M-9

ABSTRACT

This research effort was undertaken to investigate a methodology for determining the most critical elements on a fighter-bomber airbase with respect to sorties generated over a three-day period. The methodology is founded on a user definable computer simulation model written in SLAM (FORTRAN based) and supported by several FORTRAN routines. The remainder ~~of the methodology~~ entails the use of factorial experimental designs for examining airfield element criticality. The airfield elements are the experimental factors. They are set to user specified levels according to the experimental design. The model produces a single response variable --sorties generated over a three-day period. Results are analyzed with common statistical techniques (Method of Contrasts, ANOVA, Duncan's Multiple Range Test). Special attention was placed on documentation of the model to insure ease of implementation by a user. Model usage is demonstrated with two experiments and their analysis. Because this methodology does not require Monte Carlo simulation of damage to the airfield, the determination of element criticality is straightforward. The lucrative targets on the airfield are then the most critical elements which can be effectively attacked with available weapons and delivery systems.

A USER DEFINABLE SLAM AIRFIELD MODEL DESIGNED FOR EXPERIMENTATION AND ANALYSIS

I. INTRODUCTION

Background

In the spring of 1981, the Tactical Air Warfare Center (TAWC), Future Plans Division (TXP), requested that the Air Force Institute of Technology study the problem of airfield attack. The thrust of the study requirement was further defined in a subsequent visit to TAWC during the summer of 1981. The basic goals of the study were established as an analysis of airfields using computer simulation to determine which elements on an airfield are the most lucrative to attack. Airfield elements were to be viewed not only in terms of their criticality, but also in terms of their vulnerability.

After a considerable amount of research, and visits to the Airfield Attack System Project Office (SPO) at Eglin AFB, and the Joint Studies Group (TAC) at Nellis AFB, Nevada, the problems of analyzing airfield attack were more clearly defined. The desire in airfield attack, as in any other mission area, is to conduct the most effective attack possible. Many factors have an influence on whether an attack is effective when a target is large and complex, as is an airfield. The airfield attack problem is compounded even further when viewing an airfield target system which is composed of individual airfields. Totally demolishing one airfield is probably not nearly so effective as knocking out one or two very critical airfield elements on all the airfields in the target system.

Other factors which have an influence on effectiveness include: (1) delivery systems available for the mission; (2) ordnance available and appropriate for the elements to be struck; (3) delivery systems' accuracies; and (4) tactics which will be employed because of the high threat nature of airfield defenses. Tactics are dictated by the defenses for a given delivery system with a specified ordnance load. Ordnance required is dictated by the nature of the target element. Delivery accuracy is a function of the tactics the delivery system, and the ordnance. Assessment of attack effectiveness must include some consideration of the probable attrition rate which will be suffered in the attack. It can easily be seen that airfield attack is not a trivial problem.

Airfields have been targeted many times in the history of aerial warfare. One appealing choice of a target on an airfield has always been the runway. Unfortunately, with today's delivery systems, using weapons currently in the inventory, and the low altitude tactics forced by high threat defenses, the runway is extremely difficult to close. In addition, rapid runway repair capability is practiced by all major air forces (Hokanson, 1975:260-262).

Traditionally, another target element of choice has been aircraft. In today's world however, the sheltering, dispersal, and concealment of aircraft is routinely practiced in the European theater and elsewhere. These practices degrade the present day capability to directly destroy aircraft on the ground--especially sheltered aircraft. Aircraft which are not in shelters are vulnerable to attack if their location is known. Occasionally, aircraft may be located by technical means before an attack. If the dispersal areas are known, area munitions (Cluster Bomb Units which scatter their submunitions in a desired area) can be employed relatively effectively. The problem today is the trade-off between the risk to the attacker, which is high, and his relative effectiveness, which may be very low. This trade-off must be considered. It dictates that the elements of an airfield which are struck must be critical elements in order to justify the risk of exposure.

Another facet of the airfield attack problem has been determining the effectiveness of attacks. If all takeoff/landing surfaces on an airfield are rendered unusable, an attack can be called effective. But if repairs can open the runway in two hours, has the attack been truly effective? It is apparent that any useful measure of effectiveness must be able to capture the effect of runway closure, as well as other relevant occurrences, such as loss of aircraft to malfunction, or enemy action. Global measures of effectiveness for airfields usually include some form of sortie production rate. This measure can include all the effects of degraded airfield elements in a single response variable.

Other simulation studies of airfield systems have used sorties flown per day, or sorties flown over a longer time period, as their measure of merit. Sorties flown over a given time period is a very attractive response variable, not only because it globally captures all the vagaries of airfield operations, but also because it is usually the absolute measure used to judge the effectiveness of flying units. This kind of measure is also a very real and understandable representation of the amount of force which can be brought to bear over a given time frame. With the above in mind, a statement of a problem for a research effort was defined.

Problem Statement

Conduct a computer based simulation analysis of fighter-bomber airfields to determine the most critical elements of the airfields with respect to sortie generation. The most critical elements are defined as those which have the greatest marginal contribution to sortie generation. The most lucrative elements for attack are those most critical elements for which a destructive capability is available.

Problem Development

The use of a computer simulation model coupled with experimental design and analysis techniques can capture the diverse interactions of a system as complex as an airfield. However, a methodology for determining the critical target elements would have to be developed. A replicable methodology was required so that the experiment could be repeated when changes occurred in the nature of airfield systems which could alter the criticality of the individual target elements. This methodology was determined to be a computer simulation model coupled with factorial experimental designs.

A thorough search for existing airfield models ensued. The Rand Corporation models reviewed were AIDA, TSARINA, and TSAR. AIDA is an airbase damage assessment model; however, AIDA does not consider target element criticality, nor does it run over time (Emerson, 1976:1-7). TSARINA was developed from AIDA and is a more sophisticated damage assessment model, which produces output which is formatted to be used as input for the TSAR model. TSAR (Theater Simulation of Airbase Resources) uses the TSARINA inputs to assess the impact of airbase damage on sortie generation capability at an airbase, or set of airbases in a theater (Emerson, 1980:i-iv). TSAR was designed for use in evaluating proposals for improving the sortie generation capabilities of United States Air Forces Europe (USAFE) airbases.

The Kearney, Inc., AIRBASE model was built for the Air Force Armament Laboratory at Eglin AFB, Florida. The model simulates airbase operations and measures the capability to supply sorties under various user defined airbase states of damage, damage repair and resupply. The damage states are defined by changing numbers of servers and/or changing probability distribution parameters (AIRBASE, 1978).

The PHOENIX model, built by Joint Studies Group (TAC), was never released for study (Southard, 1981).

The AIDA and TSARINA models were not required for this study because damage levels did not have to be stochastically determined and analyzed for the methodology of this study. The TSAR model was structurally unsuited, and very much too large and complex to be used in the time available for the study (Emerson, 1980:1-5).

AIRBASE offered many options, and had been used successfully by Grumman in a study (Grumman working paper, 1981); however, aircraft maintenance failure was not adequately addressed in the model. In addition, AIRBASE is written totally in FORTRAN. A FORTRAN model is normally not as easy for a new user to adapt to as a simulation language model because of the lack of standard structural model graphics. There are models which are exceptions, but AIRBASE was not one of them. The inputs to the AIRBASE model are also a drawback, as they consist of a hodge-podge of formatted key punch operations (AFFDL Report 79-3018, 1978). In contrast, simulation languages like Q-GERT and SLAM contain graphical tools to construct and illustrate a structural model. The Q-GERT and SLAM models may be supported by FORTRAN routines to increase network flexibility, but the basic model remains Q-GERT or SLAM no matter how much FORTRAN support is added.

At this point in the research, the only reasonable model available was AIRBASE, and it would require extensive modification to perform the experiments desired.

Research for this study effort also included investigating languages designed for simulation--especially Q-GERT (Pritsker, 1979) and SLAM (Pritsker & Pegden, 1979). Simulation languages are designed to make it easier to understand how a model is conceptualized by including an explicit structural model. Additionally, their code is relatively easy to verify. These features are not present in a FORTRAN model. In order to use an existing FORTRAN model, one must analyze the code intimately to insure that it performs as described by the available documentation and that conceptually the model does what is desired for the given study. In most circumstances, the documentation --no matter how extensive--is never complete enough to fully understand the model without dissecting the code. Due to the factors presented above, along with the time required to decipher the AIRBASE model and modify it, the use of an existing model was a very unattractive alternative.

It became apparent a model would have to be built from scratch. This appeared to be an intrinsically worthwhile project in and of itself for four reasons. First, the learning experience of modeling is worthwhile in an academic program, and many believe that the understanding gained is the chief value of modeling (Shannon, 1975:5-7; Hoeber, 1981:4-6). Secondly, no airfield model written in a simulation language was available in the defense community. The present effort could provide a jumping off point for such a model. The third reason was that the methodology of this study, factorial experimentation on levels of airfield elements, was a new approach and no model could be found which was conveniently oriented for this approach. The last reason is the trivial case. Without a model, there was no airfield to experiment upon. To learn more about modeling in general, and Q-GERT in particular, a preliminary airfield model was constructed.

Preliminary Modeling Exercise

To help define the boundaries of the airfield study effort and to practice the conceptualization of a complex system, a preliminary model was built using the Q-GERT simulation language (Pritsker, 1979). Because airfields are such complex systems, and because of their multiple layers of interaction with the outside world, it was felt this preliminary project would help to learn both about modeling and about airfield models. The preliminary project was a very valuable experience. It was an excellent introduction to a simulation language which uses graphical symbols to construct a network model from which Q-GERT coding can be directly written. The conceptualization, computerization, and experimentation exercises were very instructive and clearly revealed the problems extant in trying to model a complex, real-world system. Q-GERT, however, did not prove to have enough flexibility, and the model developed for this analysis was written in SLAM (Pritsker & Pegden, 1979). SLAM is a language which was developed as a major advancement by the author of Q-GERT. The languages have many similarities, with SLAM being more advanced, more capable, and more flexible.

The Q-GERT exercise clearly showed that the problem would have to be limited carefully by explicit and reasonable assumptions. However, it also showed that careful modular construction of a simulation model could allow for growth, user definability, and ease of changing parameters for experimentation. These features were intentionally included in the SLAM modeling effort to facilitate experimentation, to appeal to other possible users, and to encourage change or growth for future experimentation.

Objectives

In order to address the problem of determining which elements on an airfield are most critical to sortie generation, the following objectives were pursued:

1. Develop, verify, and validate a computer simulation model to measure a response variable--effective sorties generated over a three-day period. Include the following features in the model:

   (A) Conceptual Simplicity - Use a network structural model to the maximum extent possible.

   (B) User Definability - Variables should allow for easy definition of airfield size and composition, and aircraft type and characteristics.

   (C) User Friendliness - The model should be well commented, easy to define and initialize, and it should have a user's guidance section.

   (D) Ease of Growth and Change - The structural model should have flexibility, and the documentation should allow a knowledgeable user to modify the model successfully.

2. Exercise the model and conduct experimentation in order to:

   (A) Demonstrate Use of the Model - Show how it is run, as well as the various outputs and their uses.

   (B) Conduct Sensitivity Analysis - Explore the bounds of variables external to the airfield system to test their effect on the response variable.

   (C) Screen Factors (Airfield Elements) for Criticality - Use a fractional factorial experimental design and analysis to identify the most important factors.

   (D) Identify the Most Critical Factor(s) - Use a full factorial experiment on the important factors identified in the screening experiment to isolate the most important factor(s) and/or determine a relative ranking.

   (E) Insure the Methodology Is Explicit - Detail all actions to sufficient depth so that the experiment can be replicated to verify the methodology or to analyze airfield system changes.

Structure of the Computer Model

Introduction

A discussion of fighter airfield operations is contained in Chapter II, and the simulation model is discussed in Chapter III. This section is designed to give an overview of the structure of the computer model as a lead-in to the remainder of this thesis.

The model is a network model, with discrete event orientation, written in SLAM (Pritsker & Pegden, 1979) and supported by SLAM-provided and author-provided FORTRAN support routines. The network, discrete event routines, and network support routines are interfaced by the SLAM processor, as illustrated in Figure 1.1. The SLAM input statements are read in and executed by the SLAM processor to exercise the network structure as directed, using the network and event oriented FORTRAN routines as required during the simulation. At the completion of a simulation, the response variable is printed out. The FORTRAN coding provides for: (1) initialization, (2) complex functions required in network routing and activities, (3) subroutines initiated by discrete event occurrences, and (4) user defined output.

The SLAM Structural Model

The SLAM structural model is composed of a network structure with discrete event orientation used where required. The SLAM language is composed of graphical symbols used to create structural models, and a coding language which describes the graphical symbols to the SLAM processor. The SLAM processor is FORTRAN-based and can accept extensive numbers and types of user written FORTRAN routines to embellish the SLAM structural model. The SLAM structural model is presented in Appendix A.

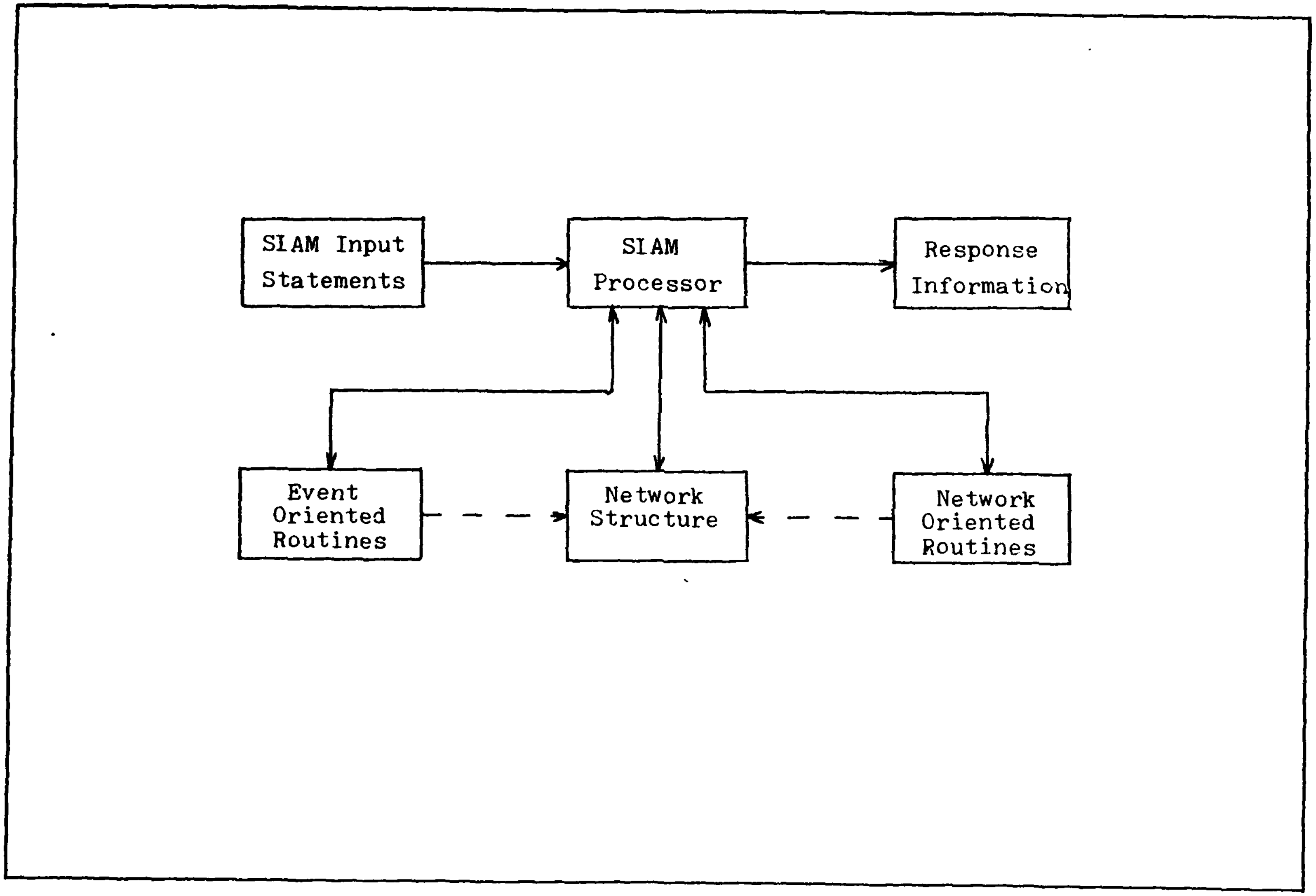


Fig. 1.1 Overview of Computer Model

The structural model was designed to keep all practicable elements and activities explicitly on the network. This effort was aimed at aiding an interested reader in gaining a more rapid understanding of the entire model through study of the SLAM structural (graphical) model.

Figure 1.2 provides an overview of the structural model. After generation, aircraft and pilots conceptually proceed in a large circle. First, they go through launch activities to fly a mission as a flight, and then recover at home base. After recovery, the aircraft are turned around for another mission, unless maintenance is required. Any time maintenance is required, it is performed before the aircraft goes to turnaround servicing. Aircraft aborting during launch are sent directly to maintenance.

The FORTRAN Support Routines

The FORTRAN code is composed of five basic sections. Three of the sections are used to support discrete event orientation and the other two support the network. All of these sections are under the umbrella of a short main program which establishes core memory requirements. The organization of these routines is shown in Figure 1.3.

Network Oriented Routines. The network oriented routines provide for initialization and network support as shown in Figure 1.3. Initialization routines are used to define the airfield composition, the scenario, the aircraft, and all the service times and probabilities which allow the

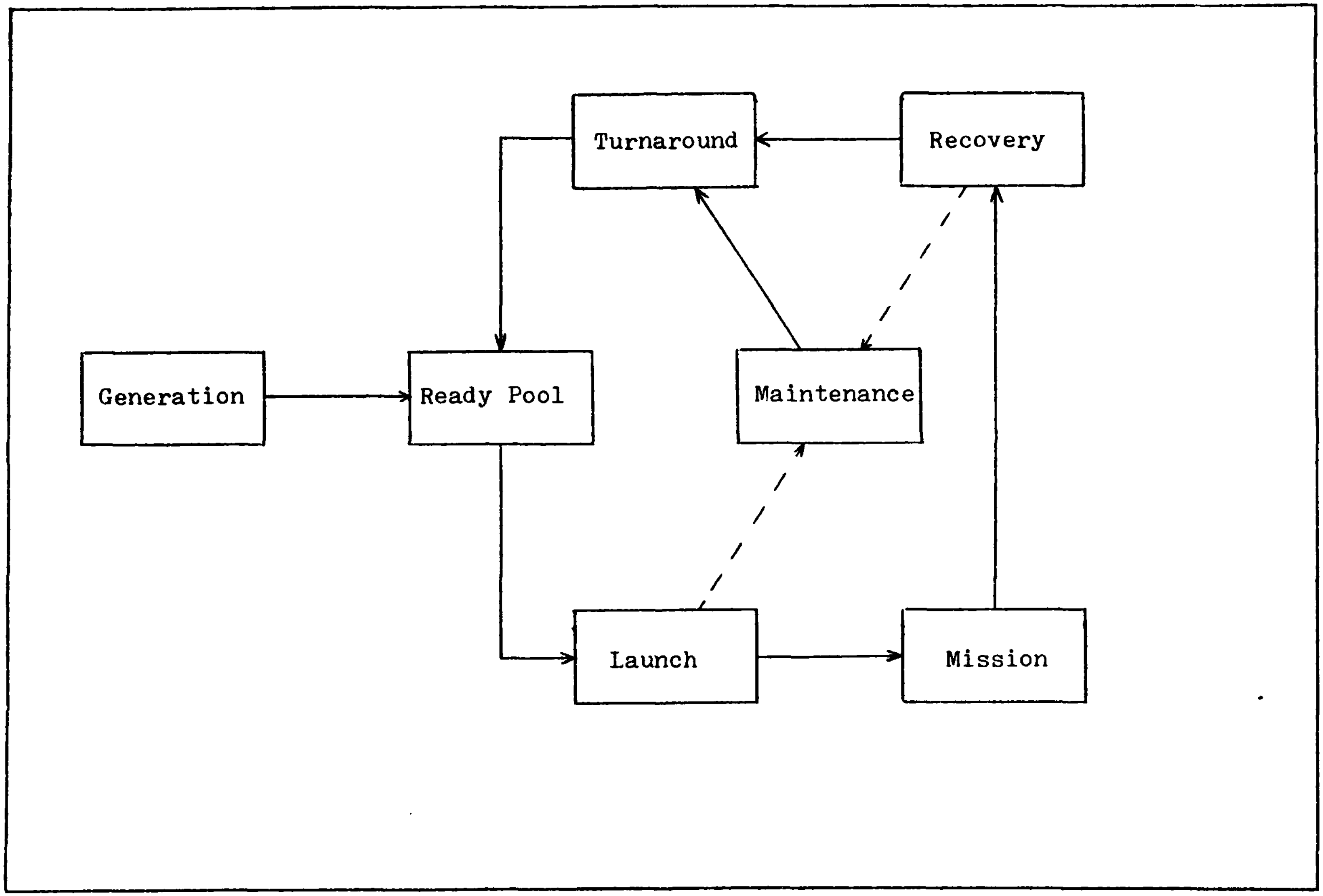


Fig. 1.2 Overview of Structural Model

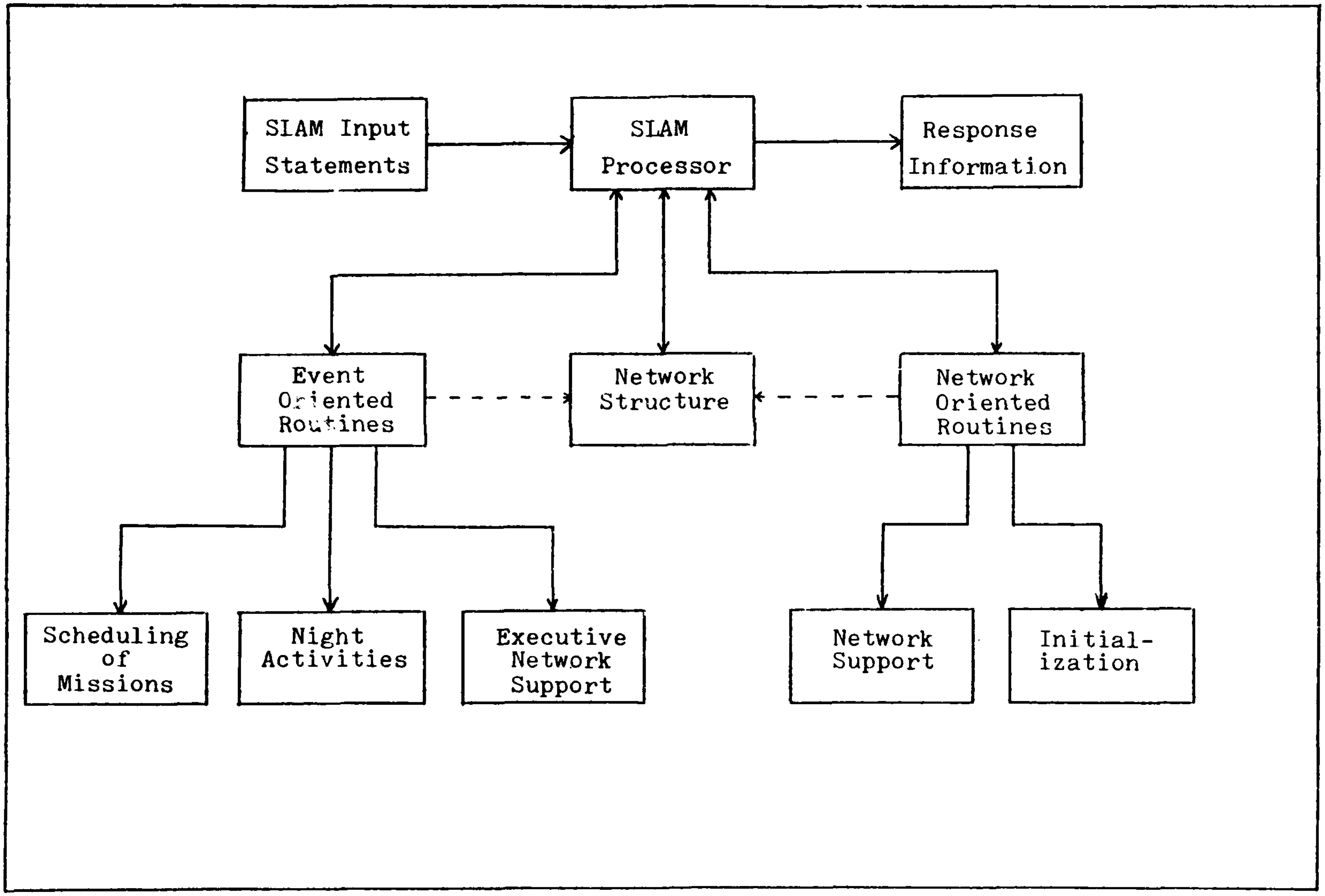


Fig. 1.3 Organization of FORTRAN Support Routines

model to function stochastically. These sections allow the user to define all the major elements of the airfield. The network support routines are individual functions built to do one simple task. They are called from the network whenever their particular task needs to be performed. These sections are fully described in Chapter III and the appendices.

Event Oriented Routines. The event oriented FORTRAN routines basically support the SLAM Executive Network, which is the controller of the major simulation activities. Other event oriented functions will be covered later (preflight spares). The major activities are shown in Figure 1.3. The scheduling of missions is a daytime routine during simulation. All flying ends in the evening. The night routines bed down the airfield and prepare for the following day's activities. All these routines are fully covered in Chapter III and the appendices.

Scope of the Model

To limit the scope of the model, certain simplifying assumptions were required. The assumptions did not limit the usefulness of the model, but they greatly reduced the size of the structural model required. The airfield was assumed to be in visual meteorological conditions (VMC) at all times. A three-day period was selected as reasonable to monitor the model's response variable - sorties generated. This period was long enough to allow start-up conditions to dampen out, yet short enough to allow reasonable computer run times.

The basic model can be tailored for up to two wings of up to three squadrons each. The aircraft travel through a network structure to perform their missions. All the basic functional activities of a fighter-bomber airfield were included in the model. Chapter II contains a general description of fighter-bomber airfields and their activities. These activities include aircraft generation, launch, recovery, turnaround servicing, and maintenance. The model is easily initialized for a desired scenario. The variables which are generally of interest for experimentation and sensitivity analysis are located in the initialization sections of the model. Experiments can be conducted by measuring the changes in the response variable caused by changes in the independent variables. The independent variables (airfield elements) are commonly referred to as factors in the experiments.

Using this approach, one could study such diverse questions as:

(1) What are the effects of a change in the percentage of pilots qualified for Quick Reaction Alert (QRA) and Flight Leader?

(2) What are the effects of changing the given Mean Time Between Failure (MTBF) for a single system, or the MTBFs of all six conceptual aircraft systems?

(3) Given a fixed scenario, what are the most critical elements of an airfield with respect to sortie generation over a three-day period?

Chapter Summary and Thesis Overview

To synthesize the information presented in the previous three sections, refer again to Figure 1.3. The model is executed by the SLAM processor, which controls the interaction of the network structure contained in the SLAM input statements, with the event/network oriented FORTRAN support routines. These interactions, under the control of the SLAM processor throughout the simulation, result in the response information contained in the model's output.

The results of the research, modeling, experimentation, and analysis are presented in the remainder of Volume I. Chapter II contains an explanation and overview of fighter-bomber airfield operations. The chapter makes the transition between real-world operations and how the model is structured and bounded. The response variable is also discussed in the chapter.

Chapter III contains a relatively detailed description of the model, including the major conceptual and user oriented aspects it contains. The model is broken down into five major airfield operations sections which include: Generation, Launch, Mission, Recovery and Turnaround, and finally, Maintenance. The SLAM Executive Network and event oriented FORTRAN routines are also covered.

The details of the policies which were used in the construction of the model are presented in Chapter IV. The chapter also covers the modular construction techniques and the continual verification processes which were used on the model. Efforts expended in validation of the model are included.

The experimental factors are defined in Chapter V. It also contains all the sensitivity analysis results. The designs, results, and analyses for the fractional factorial screening and the full factorial experiment are also included.

Chapter VI contains the conclusions reached from the experiments with the model, and from working with the methodology to establish the criticality of airfield elements. In addition, the chapter contains several recommendations for further study.

Volume II consists of three appendices. Appendix A contains the SLAM structural model and the SLAM coding. The structural model is sequenced in with the coding, subsection by subsection, and presents the clearest idea of the network functions of the model. Appendix B contains the FORTRAN coding. Both of these appendices are well commented and Appendix A is especially useful in understanding how the model functions. Appendix C contains notes to users which supplement the contents of Chapter III and Appendices A and B.

Volume III contains Annex A, which is classified. Contact AFIT/ENA, WPAFB, OH 45433, Autovon 785-5533 for information on access.

II. A FIGHTER AIRFIELD SYSTEM

Introduction

This chapter is designed to provide a transition from real-world fighter airfield operations to the airfield model. First, a brief overview of the nature of high stress surge operations is presented, followed by a synopsis of fighter airfield operations. The response variable for the model and the airfield system boundaries are presented, followed by a synopsis of the operation of the airfield structural model.

Surge Conditions

During periods of conflict and during peacetime surge exercises, a maximum effort is concentrated on flying the most sorties possible over the period of the surge. Typically, surge operations are conducted so that the tempo is quickly raised to whatever maximum sortie rate can be achieved. From that point, the tempo is decreased as necessary to avoid flying the aircraft absolutely into the ground before the surge slows to a sustainable steady-state condition (refer to Figure 2.1). If a base is surged to the point where the aircraft are falling apart, reasonable steady-state operations are normally not reattained until the base is allowed to close down (for all practical purposes) and repair aircraft.

The pace of operations normally is planned around a

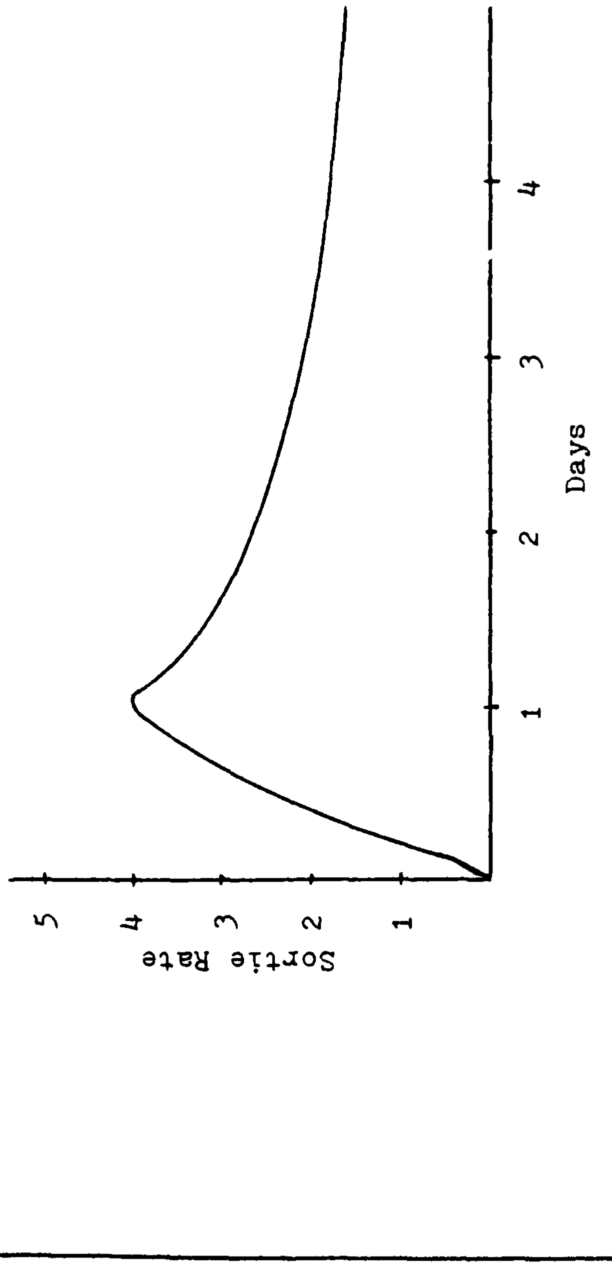


Fig. 2.1 Typical Surge Operation

higher headquarters (HHQ) mandated sortie rate. This planned sortie rate is important for many reasons. All logistics support for the base is derived from this planning factor. Resupply of expendables/consumables is based on rates of consumption derived using the planned sortie rate. These items range from such things as aircraft spare parts to bombs, bullets, and petroleum/oil/lubricants (POL). The bench stock spare parts on hand are nominally stocked, based on planned sortie rate and historical failure rates. It can easily be seen how important the planned sortie rate can become.

Most commanders strive to achieve a sortie rate in excess of that required by the planning factor. Usually, they succeed. However, in order to hedge against failure, most commanders plan an ensuing day's schedule of operations so that they can turn around their operational aircraft in a flow which experience has shown tends to maximize the effectiveness of the airfield's turnaround activities.

For day-only fighter operations, the first step in this scheduling process is to insure that HHQ required missions can be flown with a high degree of certainty. Then the remaining scheduling is done to heuristically achieve a favorable pattern of flying and turnaround factored around the HHQ requirements.

During the conduct of scheduled flying operations, the success and failure rate of missions are tracked by a command post, along with the maintenance status of all aircraft. When it is apparent that the schedule can be completely met with aircraft currently available, the remaining aircraft and any others which can be turned around are used to launch as many additional missions as possible. Twenty-four hour operations differ from the above pattern, in that the maximum effort is normally flown from late afternoon into the evening. This leaves the night period as lightly scheduled as possible, so that the aircraft can be repaired and prepared for the following day's operations. This model deals with day VFR/VMC (clear weather) fighter operations, and the twenty-four hour considerations do not apply.

There is usually prior knowledge of when a surge operation will occur. This knowledge can come from peacetime exercise notification or from intelligence Indications and Warning in a conflict situation. In any event, twelve hours' warning is usually sufficient for a fighter wing to prepare the bulk of its aircraft to begin a surge.

Fighter Airfield Operations

The conduct of fighter operations involves a repetitive series of tasks. Once the aircraft have been generated--repaired (if required), armed, and preflighted by maintenance--the cycle begins. In normal surge operations, pilots are assigned to similarly configured aircraft within a squadron. After an appropriate period of mission planning and briefing, the pilots proceed to their respective aircraft.

Pilots normally perform a walk-around inspection prior to enplaning and strapping in, after which they do a pre-start check. Next the engine(s) is started and after-start checks are accomplished. Coordination is effected between aircraft in some manner (radio, hand signals, runner) so that all flight members know when to taxi, so they can efficiently arrive in the arming area together. If an aircraft is aborted, up to the time the aircraft start engines, a spare aircraft is usually provided if available. This procedure does not delay the remainder of the flight unduly during high tempo surge conditions, and it helps get a larger number of optimally-sized flights into the air. Any other delays that occur are handled at the discretion of the flight leader and the command post. All procedures up to this point require the aid of a crew chief and perhaps an assistant.

When the aircraft arrive in the arming area, the ordnance is armed by an arming crew. In some cases, arming may be done in the parking spot to expedite launch. Arming in the parking area is usually only allowed when a malfunction of forward firing ordnance (cannon, rockets, missiles) poses no threat to anything on the airfield. In any case, once aircraft are armed, the flight leader secures clearance onto the runway for takeoff. Recovering aircraft normally have priority for the runway, depending on their fuel state. Delays are handled by the mutual discretion of the flight leader and the command post.

Once on the runway with pre-takeoff checks complete, the takeoffs proceed. In most cases, aircraft roll individually when loaded with ordnance. A small amount of time is allowed between brake releases on each aircraft (nominally

10 seconds). In the event of a malfunction on takeoff, this procedure provides a little higher margin of safety. Once airborne, the flight rejoins to an appropriate formation and proceeds on the mission. Any problems encountered in this process are handled at the discretion of the flight leader and, if time allows, the command post.

While on the mission, the aircraft are exposed to a variety of hazards, including aircraft malfunction and enemy action. In rare instances, pilots may even accidentally fly into the ground. Normally, aircraft proceed to a specified target and deliver their ordnance. Some targets may be strafed--if they are vulnerable to strafe and the defenses are not prohibitive. Ordnance is subject to malfunction. Bombs may hang up, rockets/missiles may fail to fire, or a cannon may explode or run away. Ordnance is rarely brought back to the airfield in wartime. It is jettisoned as a preferable option. Most often, it is jettisoned armed. After the target is struck, aircraft return to base, and when clearance is granted they land. The condition of the aircraft will vary from no malfunctions at all, through aircraft and/or ordnance malfunctions, to severe but flyable battle damage. Some aircraft will be lost to enemy action (attrition) or to a severe maintenance malfunction (crash), or to a combination of both.

After landing, the aircraft roll out and then taxi to a dearming area, where a dearming crew makes any remaining ordnance safe--especially forward firing ordnance. Aircraft may blow tires on landing, or otherwise malfunction, so that they must be towed rather than taxi under their own power to the dearming area. In this case, the engine is shut down and the pilot deplanes so that a tug can tow the aircraft.

Once dearmed, aircraft proceed to their squadron area for parking. Preferred parking is in an aircraft shelter. The next choice is in a dispersed revetment. The other alternative is dispersal, and possibly concealment, with no hard protection. Enroute to parking, aircraft may refuel in a Hot Pit refueling area. These areas contain fuel hydrants where refueling can be accomplished while the aircraft engine(s) is running. The use of Hot Pit refueling expedites the refueling process. If an aircraft parks in a shelter, it will normally be refueled in the shelter, either by a truck or by a pipeline system. Once aircraft reach their parking spot, the engine is shut down and the pilot deplanes.

If an aircraft lands with a major malfunction, an attempt will be made to park it in the Wing Maintenance area. If no space is available there, it will proceed to squadron parking. Aircraft requiring maintenance are scheduled for repair depending on their malfunctions. If squadron maintenance can make the repairs, they will be handled at squadron level. If not, repairs will be directed by Maintenance Control. The aircraft will be towed to wing when space is available, or wing specialists will repair the aircraft at its parking spot in the squadron area.

Aircraft with no malfunctions, or only flyable discrepancies, will be turned around for another mission.

Turnaround service is usually begun immediately after engine shutdown and is rarely hampered by a cursory maintenance post-flight and traditional pilot walk-around. Turnaround service normally consists of rearming, refueling, and a mix of maintenance post-flight and pre-flight activities. When turnaround service is completed, the aircraft joins the pool of ready aircraft. This pool is constantly being monitored by the command post and Maintenance Control in order to form and launch additional flights.

Response Variable

The basic measure of merit produced by the airfield model is effective sorties generated over a three-day period. In the computer model, this value is recorded in *SLAM global* variable XX(94). This type of measure is not only a major item of interest in a production oriented system, but also an indicator which captures all the vagaries which exist in the airfield system. An effective sortie is counted when an aircraft gets airborne, even if it later air aborts or crashes. This may appear to be slanted in favor of maintenance in the age-old battle between maintenance and operations over what constitutes an effective sortie. Actually, it stems from the fact that to determine criticality for generation of sorties, only the internal mechanism of the system is important. Once the aircraft leaves the ground, the internal mechanism has fulfilled its function.

Each individual run of the model will produce a value of the response variable (number of sorties generated over a three-day period), which is a data point for use in experimentation and analysis.

System Boundaries

The airfield is a system which responds to many external influences. Many of these influences are inputs to the airfield model. As was stated above, the basic measure of merit which the model calculates is effective sorties flown over a three-day period. In an analysis aimed at identifying the most critical factors in the generation and turnaround cycle, it is important to isolate those factors which are internal to the system. Since the only concern is for the ability of the airfield to put aircraft into the air, the external factors merely have to be set at reasonable levels so they do not have any undue influence on the response variable.

For the above reasons, only those activities and facilities which play a direct part in the generation, launch, recovery, turnaround, and maintenance of aircraft are considered to be internal to the system. They are the factors for experimentation. All other factors are external to the system and will be set to fixed reasonable levels. This approach isolates the airfield so it may be studied.

Airfield Elements of the Model--Functions and Relationships

A typical fighter airfield can be conceptually illustrated with a diagram such as Figure 2.2. In most cases,

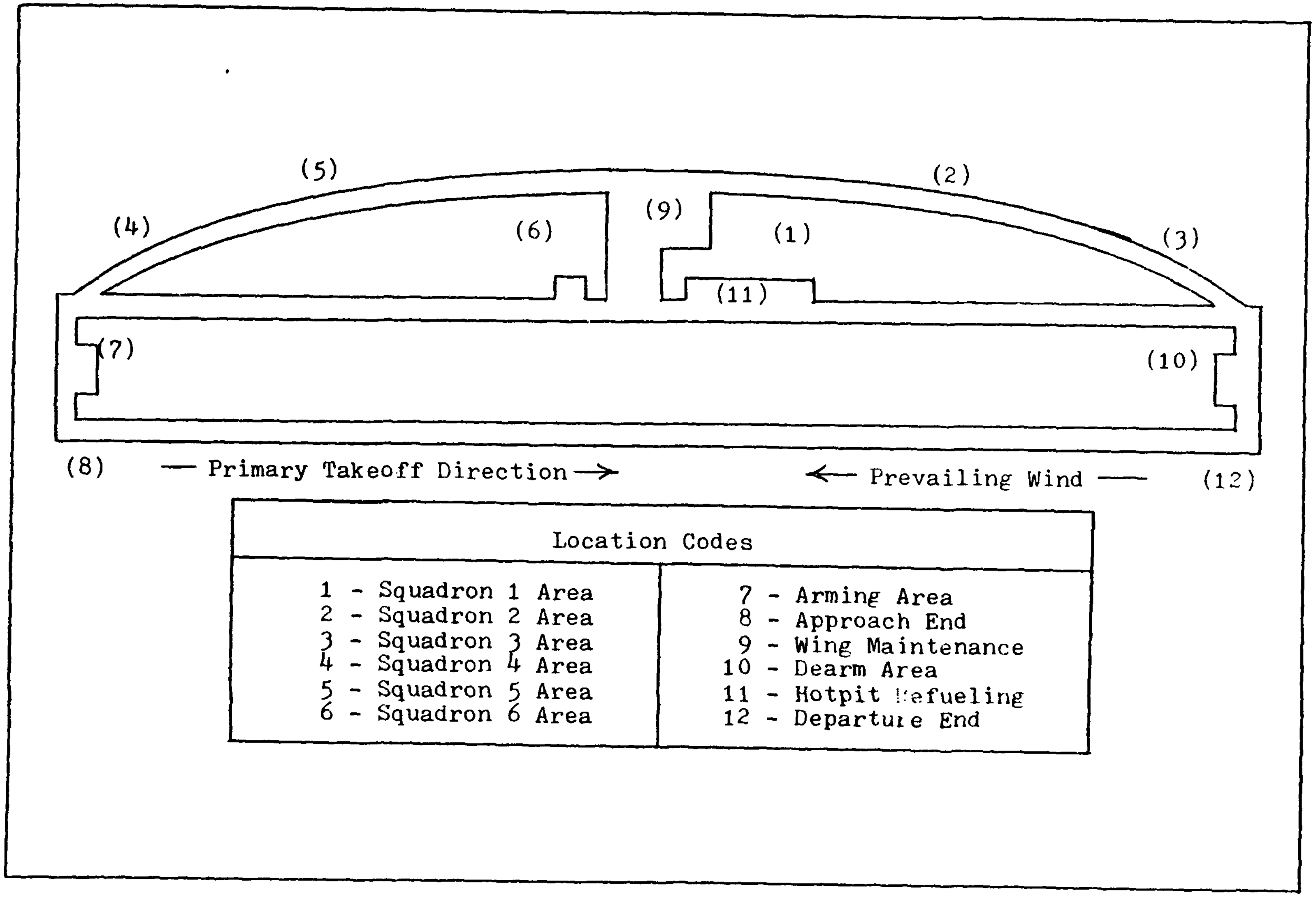


Fig. 2.2 Typical Fighter Airfield

squadron parking areas are discretely defined. Depending upon the taxi route an aircraft takes just prior to engine shutdown, a Hot Pit refueling facility may be convenient for some squadron's aircraft.

Airfields are constructed so that the runways are aligned with the local prevailing winds. This results in a primary takeoff direction during a given season as illustrated in Figure 2.2. To launch on a mission, aircraft proceed from the squadron area to the approach end of the active runway in order to marshall and arm. After arming and checks are completed, the aircraft obtain clearance onto the active runway and takeoff.

For recovery, aircraft will land on the approach end and roll out on the runway. At the departure end of the runway, the aircraft turn off the active and park in the dearm area to have their remaining ordnance safed. After dearming, aircraft will taxi as directed and required. Aircraft which abort during the launch process, and those taxiing back after a mission, will Hot Pit refuel if it is convenient to their taxi route and if it will not interfere with any repairs that the aircraft may require.

A conceptual overview of a fighter airfield structural model is presented in Figure 2.3. Aircraft and pilots are generated (created and initialized) and placed in separate ready pools. Some aircraft and pilots must be on Quick Reaction Alert (QRA). Some aircraft are still in maintenance at the beginning of the surge.

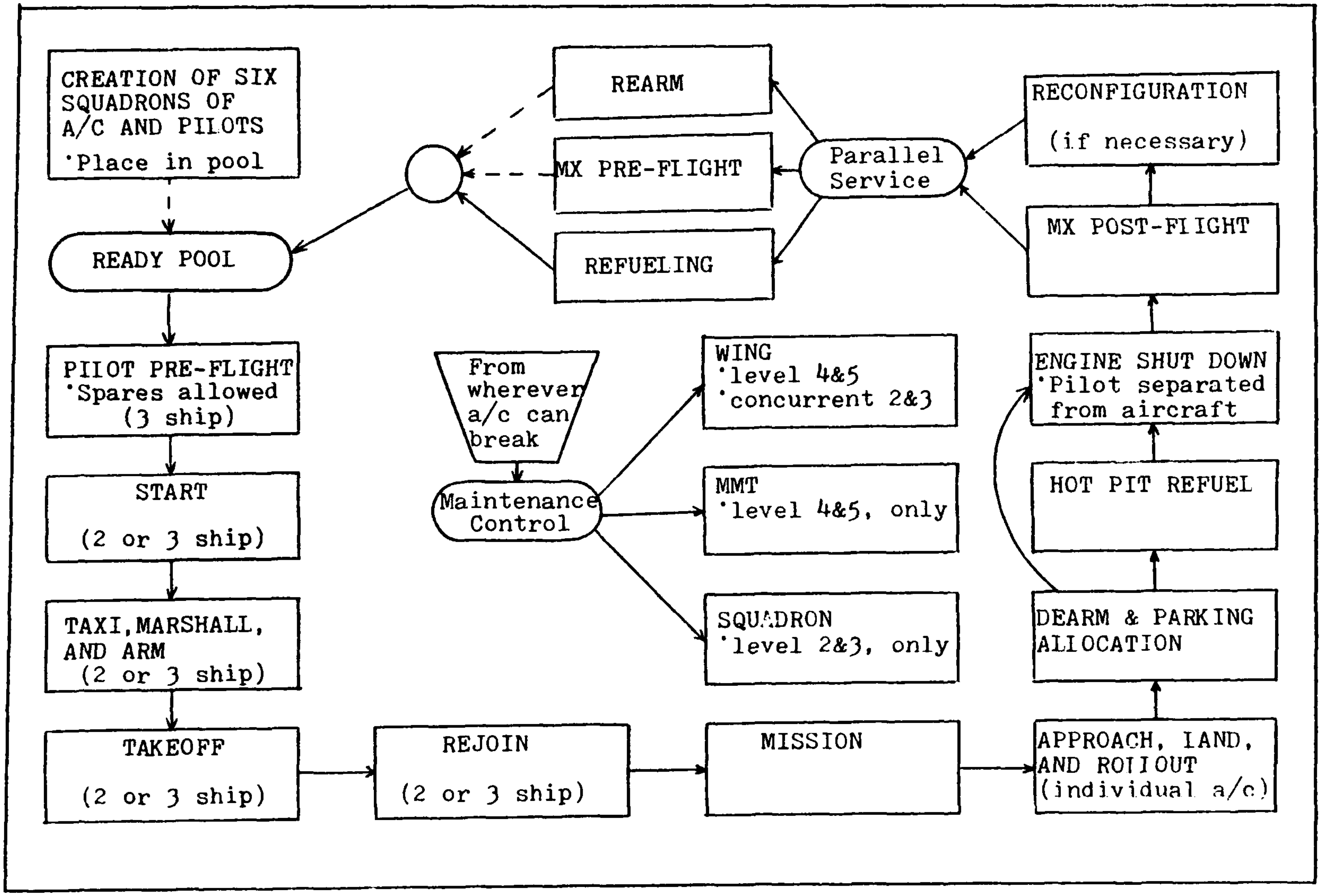


Fig 2.3 Overview of Airfield Structural Model

The launch process begins with the Scheduler forming a flight of aircraft with pilots from a common squadron. The pilots preflight their aircraft. If any aircraft are aborted, spares are allowed at preflight. Next, engines are started and the flight taxis, marshalls, and arms. At each activity, aircraft may abort due to maintenance failure. As long as at least two aircraft remain in the flight with at least one flight lead, the flight may proceed. After arming, the flight takes off, rejoins, and proceeds on the mission. Aborted aircraft may proceed to maintenance from any point in the launch process.

During the mission, aircraft may be shot down or crash due to maintenance failure and/or battle damage. Aircraft deliver their ordnance and possibly experience weapons release malfunctions. External fuel tanks may be jettisoned as part of the mission profile or in response to being attacked by another aircraft. After the mission, aircraft approach, land, and roll out on the active runway. Aircraft are then taxied or towed, as required, to dearming and then to parking. Taxiing aircraft may Hot Pit refuel, if convenient. Some aircraft may be parked at Wing Maintenance for service if they have serious malfunctions and if space is available. Wing Maintenance shop resources are conceptually viewed as capacities for aircraft. In Wing Maintenance, lesser problems are considered to have been repaired concurrently with major problems. When repairs are complete, aircraft proceed to turnaround servicing.

If no space is available at Wing Maintenance, aircraft with serious problems are repaired in the squadron area by wing specialists (MMT). Each MMT unit resource is conceptually able to work on one aircraft at a time. No work on lesser problems takes place until MMT unit repairs are completed. If an aircraft is awaiting an MMT unit for service and space becomes available at wing, the aircraft is towed to wing for service. Lesser maintenance problems are repaired by squadron maintenance shops. Squadron level repairs are performed concurrently. When all repairs are completed on an aircraft, it proceeds to turnaround servicing.

If an aircraft experiences a malfunction which requires that it be towed, the engine is shut down and the pilot is removed from the aircraft and returned to the Pilot Ready Pool for the appropriate squadron. The aircraft is towed to the appropriate maintenance facility. When repairs are completed, aircraft proceed to turnaround servicing. A complete discussion of maintenance is contained in Chapter III.

Aircraft which sympathetically abort (ground or air) and have no non-flyable discrepancies are returned to the squadron area for engine shutdown, pilot deplaning, and turnaround servicing. Aircraft returning from a mission with no non-flyable discrepancies are returned to the squadron area for engine shutdown and pilot deplaning. Their configuration is checked and external fuel tanks are hung on them, if required. When reconfiguration is completed, the aircraft enter turnaround service.

Turnaround service consists of rearming, refueling, and an abbreviated mix of maintenance post-flight and pre-flight activities. Rearming and refueling are only accomplished when required. The services are performed concurrently. When turnaround servicing is complete, the aircraft are returned to their respective squadron's Aircraft Ready Pool.

The preceding material has presented an overview of the structural model. The discussion has closed the loop in the network in which the pilots and aircraft flow. The major elements of the airfield model were covered with an overview of their functional relationships. The details of the major aspects of the model, and a more complete treatment of the network are presented in Chapter III. The remaining details are covered by the extensive comments in Appendix A and Appendix B.

III. THE SIMULATION MODEL

Introduction

This chapter provides a relatively detailed description of the model. The first section covers the major conceptual and user oriented aspects of the model. The remainder of the chapter covers the major sections of the model shown in Figure 3.1. The sections are covered starting with Creation and Initialization (Generation). Then, in order, The Launch Process, The Mission, Recovery and Turnaround Service, and Maintenance are covered. Finally, the Executive Network and the event oriented FORTRAN subroutines are explained. The chapter closes with a summary. Those readers who are well schooled in computer simulation may wish to look ahead to Figure 3.5 to see an overview of how all the separate sections of coding interact.

Major Aspects of the Model

Dynamic Maintenance Failure Code

Aircraft have been provided with six conceptual systems. The systems are:

(1) Electrical

(2) Engine/Fuel

(3) Hydraulics/Pneumatics

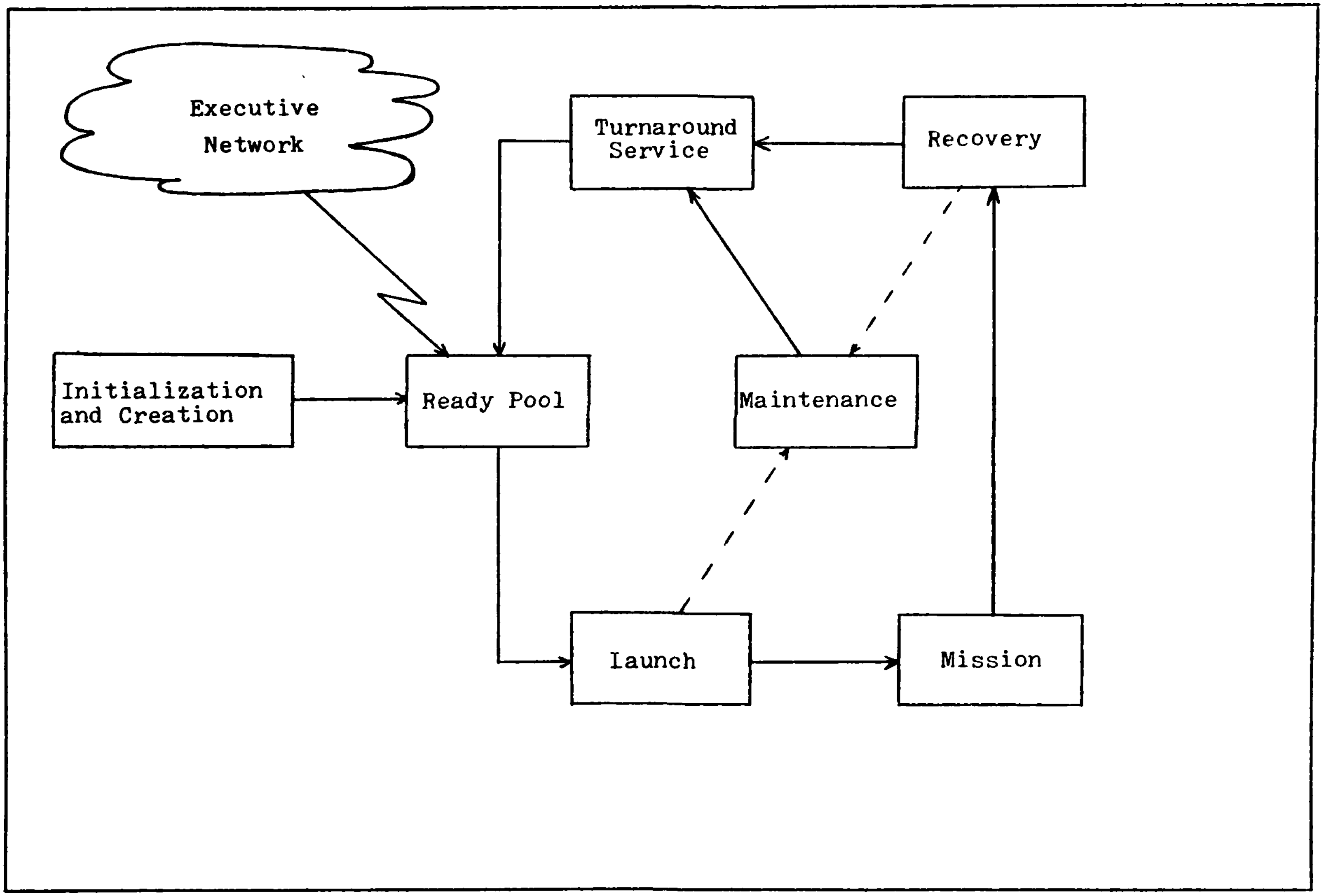


Fig. 3.1 Major Sections of the Airfield Model

(4) Airframe

(5) Comm/Nav/Instruments/Radar

(6) Fire Control/Weapons Release

Aircraft are initialized with a value for total engine run time (Attribute(7)), which is uniformly distributed between zero and 200 hours (major overhaul periodicity assumed at 200). Each system is initialized with a value for Next Time of Failure (NTOF) (Attribute(19) through Attribute(24)), which is a random pick from the Beta distribution using that particular system's Mean Time Between Failure (MTBF). Total engine run time is accumulated after each activity where the engine is running. Every time the total engine run time is updated, each system is checked to see if it has failed.

If total engine run time exceeds NTOF, then that particular system has failed. The model uses probabilities aggregated for each system to determine the level of failure. The cumulative distributions for the level of failure for each system are found in Section 5 of Subroutine INTLC. The failure level is returned by the FORTRAN function USERF(51) and the Maintenance Failure Code (Attribute(18)) is updated to reflect the change.

The Maintenance Failure Code is a six-digit number whose individual digits, in order, carry the current status (zero to five) of each aircraft system. At each place on the network where the failure code is updated, routing is provided for the aircraft to begin proceeding to the appropriate maintenance facility for repair. The failure code is checked by

USERF(37), which returns the largest value of any digit. If any digit is greater than or equal to two, the aircraft requires repair. The failure levels are:

Flyable

0 - No Discrepancies
1 - Only Flyable Discrepancies

Non-Flyable

2 - Minor Discrepancy (Short Repair Time)
3 - Minor Discrepancy (Longer Repair Time)
4 - Major Discrepancy (Short Repair Time)
5 - Major Discrepancy (Longer Repair Time)

This method of handling systems can be easily expanded to handle more systems, or the six systems may just be reconceptualized to be other systems to suit the user's needs. All that is required is a knowledge of modular arithmetic. The upshot of dynamic failure is that once the aircraft have been initialized, they proceed through the network during simulation and fail wherever they happen to be when the time values dictate a failure. If an aircraft is airborne, the failure code is compared to a matrix of standards, and if the failure is serious enough, the aircraft crashes. If the aircraft is on the ground, the failure code is checked against another matrix to see if the aircraft will have to be towed. If the aircraft is airborne and has a Battle Damage Code (Attribute(16)) value greater than zero, yet another matrix is checked to see if the aircraft should crash. (The Battle Damage Code is explained in the Mission section of this chapter.) The three matrices are in Section 1 of Subroutine INTLC. Additional information on the Maintenance Failure Code is covered in the Maintenance section of this chapter.

Subroutine INTLC

Subroutine INTLC is a SLAM provided routine which has been tailored to allow a user to define the airfield composition and the aircraft type. This routine helps to establish the scenario that will affect aircraft movement through the network.

In this subroutine, the probabilities of system failure and level of failure are specified. Repair service times are specified. The values described in the preceding section are put into the three matrices for crashing and towing. Probabilities for ordnance delivery and malfunction are specified. Battle damage occurrence and level of damage probabilities are defined. The number and type of aircraft parking spaces and the distances between facilities are specified. The transformation function for setting the Maintenance Failure Code based on the Battle Damage Code is defined. The Uniform distribution for initialization of total engine run time is specified. Rates of movement on the airfield are set. The aircraft characteristics are established. Service times are specified. Sunrise and sunset are specified, as well as certain probabilities of delay. After these variables have been specified to the user's satisfaction, the variables in Subroutine USERI should be dealt with.

Subroutine USERI

This subroutine contains many of the variables which will be of interest for experimentation, and their initialization continues the specification of the system begun in Subroutine INTLC. In this section, the user specifies planned sortie rates and the desired schedule, by area. There are three conceptual areas. Area 1 is relatively near the airfield and sorties may be flown without external tanks, or with minimum external tanks (for example, a centerline tank). Area 2 is further away and requires maximum external fuel if aerial refueling is not available. Area 3 is further than Area 2, and sorties to this area are only flown in gaggles (large groups of flights on a single mission), with maximum external fuel. For the details of scheduling, refer to Appendix C.

Configuration, mission duration, probability of attrition, and probability of tank jettison must be specified for each area. The number of squadrons, aircraft and pilots per squadron, and pilot qualification must be specified. MTBFs and the shape parameters for each system's distribution must be defined (see Appendix C). The number of resources for maintenance and refueling are specified. The number of runways is set. Also in this routine is the print statement control switch. Six levels of print statements can be specified, as well as the simulation time for throwing the switch on and off during a simulation run. When these values have been specified, the model is ready to run.

Function USERF(IFN)

This function is the yeoman of the FORTRAN coding. There are thirteen sections in USERF(IFN). The sections include:

(1) Utility
(2) Parking
(3) Examine Code
(4) Reset Code
(5) Alter Code
(6) Travel Times
(7) Turnaround
(8) Wing Service
(9) MMT Service
(10) Squadron Service
(11) Maintenance
(12) Statistics
(13) MTBF Distribution

Each of the 75 functions is individually commented. Refer to Appendix B, pages 148-194.

One feature of USERF(IFN) deserves a special note. Each function is addressed through a single access point, which then uses a COMPUTED GO TO to access the specified function in USERF(IFN). The result of this is best explained with an example. Suppose a reader encounters a function, USERF(76), in the SLAM coding, and he wishes to know more about it. It can be found at label 7600 in the FORTRAN code in Appendix B. Merely riffling through the pages of the Appendix, the label 7600 is rapidly found on page 178, and just above it the comment that explains exactly what the function does. Any function in USERF(IFN) can be found just as rapidly by adding two zeroes to the function number as was done in the example above.

Concurrent Service

Concurrent service is performed when more than one service activity is being performed on an aircraft at the same time, i.e., concurrently. Turnaround Service and Squadron

Maintenance are concurrent service activities. To accomplish this with SLAM, a copy of the aircraft entity is routed through each activity and then sent to a Queue node before a Match node. The entities then wait to match back together based on the unique tail number of the aircraft. When there is an entity in each Queue node with the same value in the attribute specified on the Match node--in this case Attribute (2) (the tail number)--the entities are allowed to flow. On the output side of the Match node, all of the copies of the aircraft entity except one are destroyed. Examples of this type of servicing may be clearly seen in the structural model in Appendix A. Refer to Turnaround Service, pages 95 and 96; and Squadron Maintenance, page 121.

Mean Time Between Failure (MTBF) Probability Distribution

Several probability distributions for MTBF were reviewed before a final selection was made. The Log Normal and Exponential were rejected because their lack of shape parameter limited their usefulness for a user definable function. Additionally, the Log Normal, Exponential, Weibull, and Gamma are all defined between zero and infinity. An upper limit of infinity is an undesirable characteristic when computing failure times. Truncation could have been used, but deciding where to truncate would have to be dealt with by every user.

As a result of the above, the Beta distribution was selected. The Beta uses shape parameters which allow for ease of user definability. The Beta is only defined between zero and one, which obviates the truncation issue.

In order to use the Beta distribution, two issues must be dealt with. First, the shape parameters have to be chosen. The shape parameters may be tailored to reflect the user's attitude about the reliability of each particular conceptual aircraft system. A discussion of this particular process is contained in Appendix C. Once the shape parameters have been chosen, the second issue, how to calculate Next Time of Failure (NTOF) for a particular system, may be resolved in a straightforward manner.

The user must have estimates of MTBFs for each of the conceptual aircraft systems. The estimates are used in the following manner. The values returned by the Beta are treated as normalized values of time between failures. The normalized expected value of the Beta (ALPHA,BETA) is:

$$\hat{E}(x) = \frac{\text{ALPHA}}{\text{ALPHA + BETA}}$$

where ALPHA and BETA are shape parameters (Law & Kelton, 1982: 165). The normalized value of time between failures is defined as:

$$T\hat{B}F = \frac{\text{MTBF}}{\text{MAXIMUM TIME TO FAILURE (MAX)}}$$

Setting them equal,

$$\hat{E}(x) = T\hat{B}F = \frac{\text{ALPHA}}{\text{ALPHA + BETA}} = \frac{\text{MTBF}}{\text{MAX}}$$

and solving for MAX,

$$\text{ALPHA} \cdot \text{MAX} = \text{MTBF} \cdot (\text{ALPHA} + \text{BETA})$$

$$\text{MAX} = \frac{\text{MTBF (ALPHA + BETA)}}{\text{ALPHA}}$$

$$\text{MAX} = \frac{(\text{ALPHA} + \text{BETA})}{\text{ALPHA}} \cdot \text{MTBF}$$

results in:

$$\text{MAX} = (1 + \frac{\text{BETA}}{\text{ALPHA}}) \cdot \text{MTBF}$$

Having solved for the maximum time of failure, the value can be used to multiply by the random draw from the Beta distribution with given ALPHA and BETA parameters to find a random value for the system life time before failure. To make this value useful in the model it is added to current total engine running time to define a particular system's Next Time of Failure (NTOF) in terms of total engine running time.

Mathematically, this process is defined for each system as:

$$f(\text{ALPHA},\text{BETA},\text{MTBF}) = \text{MAX} \cdot \text{Beta}(\text{ALPHA},\text{BETA})$$

Substituting for MAX, results in:

$$f(\text{ALPHA},\text{BETA},\text{MTBF}) = (1 + \frac{\text{BETA}}{\text{ALPHA}}) \cdot \text{MTBF} \cdot \text{Beta}(\text{ALPHA},\text{BETA})$$

It can now be clearly seen that the crucial step in this process is choosing the shape parameters. Once they are chosen, and an estimate for MTBF for each system is available, the NTOF's of the systems are easily calculated. Appendix C contains a discussion of how to choose the shape parameters.

Creation and Initialization (Generation)

Normal Squadron Processing

A simulation run commences with the creation of aircraft and pilots for each squadron the user has declared active in Subroutine USERI. The model creates 50 aircraft, allows the number the user has specified to be initialized, and then terminates the excess. The model creates 75 pilots, allows the user specified number in a squadron to be initialized, and then terminates the excess. A user selectable percentage of aircraft are initially operational. The remaining aircraft are sent to maintenance. The aircraft sent to maintenance are evenly distributed to maintenance. Each system on an aircraft has a 50 percent chance of failure in order to spread the aircraft across the spectrum of repair facilities. Failure levels of failed systems are uniformly distributed from one to five. A user specified number of aircraft and pilots are placed on Quick Reaction Alert (QRA). The number is specified in SLAM global variable XX(61).

Aircraft carry information with them in their attributes. The attributes of aircraft, pilots, and missions are listed in the SLAM coding, Appendix A, pages 3 and 4, lines 28 to 109. Aircraft are initialized with squadron number, tail number, parking spot, total engine running time (a pick from a user specified Uniform distribution), and NTOFs from the previously described Beta distributions. Aircraft are initialized with ordnance loaded because aircraft arming is a part of the generation process. Generation is accomplished prior to the start of Day 1. Aircraft proceed to their squadron's Aircraft Ready Pool and await a mission.

Pilots also have attributes. Pilots are initialized with squadrcn number, pilot identification number, and qualification (flight lead, QRA qualified, or on QRA status). When pilots enter an aircraft, the pilot attributes are transferred to aircraft attributes, and when the pilot deplanes, his attributes are transferred back from the aircraft entity to the pilot entity. After initialization, pilots who are not on QRA status are placed in their squadron's Pilot Ready Pool to await a mission.

Replacement Squadron Processing

During a simulation run, Subroutine EVENT(IEV) monitors squadron aircraft status each evening. If a squadron has fallen below a user specified number of operational aircraft, Subroutine RESUPLY (sic) schedules the arrival of a replacement squadron around midday the following day. This is accomplished by the setting of a SLAM global variable assigned to each squadron. At about 1300 on Days 2 and 3 of the simulation, the model creates 50 aircraft. If the global variable for a squadron has been set to one, the aircraft are processed into the system. If the variable equals zero, the aircraft are terminated. The aircraft are processed by adding pilots with qualifications based on the same ratios the user specified for the initial squadrons. This results in the replacement squadrons having the same relative number of flight leads, QRA qualified flight leads, and pilots, as the originally created squadrons. This is accomplished by simple counter functions. Aircraft are given user specified configurations, and all other attributes are set in the same manner as the initial aircraft. Aircraft proceed to approach and land to enter normal processing.

The Launch Process

Mission Global Variable (MGV)

Following each launch activity, the status of each aircraft in a flight must be checked to determine whether it has experienced a maintenance failure. This must be done to determine further routing for the aircraft. The check of each aircraft is accomplished using the MGV. This occurs after preflight; engine start; taxi, marshall, and arm; takeoff; and rejoin.

The value of the MGV is carried in SLAM global variable XX(II). (II) is the value of the mission number of a particular flight. Mission numbers range from one to forty-six, which is adequate to handle the maximum number of flights which can be generated on the airfield at any one time during reasonable simulation scenarios. The value of XX(II) is set to zero just prior to entering each launch activity. When an aircraft fails, the value of XX(II) is changed by adding a specific amount depending on the aircraft flight position numbers which have experienced failures. Refer to Figure 3.2 for the specific numbers which are added and the totals which can result to

Rules for Setting XX(II):

1. XX(II) initialized to zero prior to each launch activity, II is the flights mission number.

2. XX(II) set when an aircraft fails:
   if aircraft 1 fails - XX(II) = XX(II) + 2,
   if aircraft 2 fails - XX(II) = XX(II) + 4,
   if aircraft 3 fails - XX(II) = XX(II) + 5.

Aircraft Status based on Mission Global Variable				
MGV	Aircraft 1	Aircraft 2	Aircraft 3	3-Ship Only
0	okay	okay	okay	no
2	failed	okay	okay	no
4	okay	failed	okay	no
5	okay	okay	failed	yes
6	failed	failed	okay	no
7	failed	okay	failed	yes
9	okay	failed	failed	yes
11	failed	failed	failed	yes

Fig. 3.2 Mission Global Variable (MGV)

define the maintenance status of the flight.

The MGV is the critical element in the conditional routing of aircraft during the launch process. It is conceptually a simple technique. The important considerations in using the MGV are resetting it to zero, and insuring its value is set prior to the conditional branches where it is used. The timing problem is handled by adding .0001 delays to activities leading to the nodes where branching occurs. This allows each aircraft to influence the MGV value prior to the node at which branching will occur based on the value of the MGV. The .0001 delay works because the SLAM processor operates on entities sequentially. The processor takes an entity as far through the network as it can before taking a new entity to process.

Preflight Spare Procedures

When an aircraft experiences a maintenance failure during preflight, it is routed to an event node which calls Event 10. Discrete event orientation is used to enable the use of SLAM provide d Subroutines RMOVE, SCHDL, and FILEM. The Event 10 routine searches the appropriate squadron's Aircraft Ready Pool, and if an aircraft is found, RMOVE removes it from the ready pool. Next, SCHDL schedules Event 11 to be called after a short delay. The delay time is the conceptual time for a pilot to change aircraft.

Event 11 is a routine which uses FILEM to place the spare aircraft into the MXTEAM await node (PFRS), so that the spare can acquire a crew chief resource and then proceed through the network to catch up with the flight unless it experiences a failure. If the spare fails, another iteration of this process can occur. If the spare completes the preflight successfully, the flight is allowed to proceed to engine start.

Sympathetic Aborts

For the reader who is unfamiliar with fighter operations, a brief explanation of aborts will be provided. During fighter operations, a minimum number of aircraft required in a flight is specified by commanders, based on tactical considerations. In this model, the minimum number of aircraft in a flight is two. As long as there are at least two aircraft in a flight, and at least one flight lead, the flight will proceed. When the flight drops below either of those minimums, the mission is scrubbed, and an aircraft which aborted solely because of dropping below two aircraft and/or at least one flight lead is referred to as a sympathetic abort.

Sympathetic aborts can occur on the ground or in the air. As an example, consider the case of a flight which has become a two-ship and has a flight lead only in aircraft number one. If either aircraft aborts in this situation, the other becomes a sympathetic abort. Sympathetic ground aborts proceed to turnaround service. Sympathetic air aborts have a delay in the air which conceptually allows for time to burn down gas and/or jettison ordnance in order to get below maximum gross weight for landing. (Most fighter aircraft can take off much heavier than they can land, due to the stress that landing places on the landing gear and tires.) After landing, sympathetic air aborts are processed like any other landing aircraft.

Flight Lead Manipulation

As in the real world, the model requires at least one flight lead in every flight. Flights are organized by Subroutine ORGANPT according to decision rules contained in the comments in Appendix B, page 208. Three cases are defined:

	A/C #1	A/C #2	A/C #3
Case 1	FLT LEAD	PILOT	PILOT
Case 2	FLT LEAD	PILOT	FLT LEAD
Case 3	FLT LEAD	FLT LEAD	FLT LEAD

Case 2 is the preferred case. Case 2 is more robust, in that the chance of losing a mission due to a sympathetic abort is much less than for Case 1. Case 1 is preferred after Case 2, and Case 3 is least preferred because it squanders flight leads.

During preflight and three-ship start engines, it is possible for lead's aircraft to be aborted. This presents no problems for Case 2 and Case 3, or when a spare aircraft is available at preflight, in any case. However, there is a problem when the flight is Case 1 and no spare is available. When this occurs, the problem is handled explicitly on the network. The flight lead is assigned the number three aircraft. The broken number one aircraft goes to maintenance and the number three pilot returns to the ready pool. Additionally, the number three aircraft is redesignated as the number one aircraft and the flight is redesignated as a two-ship. Refer to Appendix A, page 42, for a graphical depiction and to the preceding pilot preflight SLAM coding for the explicit means of transferring the attributes of the pilots to and from the aircraft.

Takeoff Clearance

As in the real world, the number one aircraft of the flight secures clearance for takeoff in the model. Conceptually, this is modeled by having the lead aircraft await the acquisition of the runway resource before the flight can takeoff. Refer to Appendix A, page 57, for a graphical depiction. After takeoff, the lead aircraft frees the runway resource. Three-ship flights have priority over two-ships. Landing aircraft have priority over aircraft waiting to take off. Two-ships acquire the runway in the same manner as three-ships.

The Mission

Introduction

The processing of aircraft while on a mission is handled by Function USERF(15). See Appendix B, pages 153 through 156, for a listing. The FORTRAN routine determines what happens to the aircraft during the mission based on the user defined probabilities set in Subroutines INTLC and USERI.

Mission Duration

The mission duration is the same for all aircraft in a flight and the value is set in SLAM global variable XX(II). XX(II) was used as the Mission Global Variable (MGV) during launch, but it becomes mission duration while the aircraft are airborne. The mission duration is a pick from a triangular distribution defined by the user in Subroutine USERI. See Appendix B, page 132, lines 102 to 106.

Attrition

Aircraft may be shot down while on a mission. The probabilities are user defined in Subroutine USERI (Appendix B, page 132, lines 113 to 118). The probabilities are specified by the area the mission is flown to and the size of the flight, two-ship or three-ship. If an aircraft is attrited, it is routed to a file (JUNK) with Attribute(16) set to 99. See the structural model for routing details (Appendix A, page 82).

Battle Damage

Aircraft may incur battle damage while on a mission. The probabilities of incurring battle damage and the probabilities of level one through five damage are user defined in Subroutine INTLC (Appendix B, page 141, lines 484 to 494). If an aircraft is determined to have received damage, the level of the damage is carried by Attribute(16) which was initialized to zero. Aircraft which incur battle damage are checked to see if they should crash. This will be discussed in the next section.

Aircraft Failures While Airborne

Aircraft may fail while airborne. If the failure, or the combination of the failure and battle damage, is greater than user definable levels, the aircraft will crash (Appendix B, page 140, lines 430 to 458). Aircraft with only maintenance problems are compared to six-digit numbers in Data Statement LCRSH. This is done by USERF(13) (Appendix B, page 152). Aircraft with battle damage are compared to seven-digit numbers in Data Statement LBAT. This is done by USERF(14) (Appendix B, pages 152 to 153).

Ordnance

Ordnance is expended and weapons/weapons release malfunctions occur based on user specified probabilities (Appendix B, pages 140 to 141, lines 452 to 480). Ordnance malfunctions require longer service times in dearming after the aircraft lands. Malfunctions also require markedly longer service times during rearming. The service times simulate release equipment changeover, or gun maintenance/replacement. The dearming and rearming times are user definable in Subroutine INTLC (Appendix B, pages 143 to 144, lines 618 to 639).

External Fuel Tanks

Configurations are user specified by mission area. Tank uploading and downloading times are user specified. Tank jettison probabilities by mission area are user specified. All these values are inputs to Subroutine USERI (Appendix B, page 132, lines 90 to 99 and 121 to 125). The probabilities should be chosen to reflect the user's opinion of the flight profile for each area and the probability of being attacked by another aircraft.

Summary

After the above described routines have been executed, the aircraft will end up in the file for aircraft leaving the system (JUNK), or they will proceed to approach to acquire the runway resource and land. The mission routines will have defined the aircraft attributes to describe the state of the aircraft. The information carried by the attributes will allow the recovery, turnaround servicing, and maintenance portions of the model to properly process the aircraft until it is returned to the appropriate Aircraft Ready Pool.

Recovery and Turnaround Service

Approach, Landing, and Dearming

Aircraft returning from a mission proceed to approach (APPR) and await the runway resource for landing. Aircraft acquire the runway individually for landing. When the runway is acquired, the aircraft lands; and in most cases, frees the runway and taxis to dearming (DEA3). Each aircraft is checked by USERF(12) to see if an aircraft malfunction exists which requires towing (for example, a blown tire). If towing is required, the aircraft is towed to dearming (DEAR) (Appendix A, page 85).

USERF(12) is used many times in the model to determine if an aircraft which has broken requires towing. The function makes the determination by comparing the aircraft's Maintenance Failure Code against the user definable Data Statement LTOW located in Subroutine INTLC (Appendix B, page 140, lines 450 and 451). Whenever an aircraft has to be towed, the pilot is separated (PSEP) from the aircraft and returned to the appropriate Pilot Ready Pool. When aircraft are towed, they move around the airfield at a different rate than when they are taxied. Rates of movement of aircraft taxiing, aircraft being towed, and pilots in a vehicle, are set in Section 6 (Travel) of Subroutine INTLC (Appendix B, page 143, lines 590 to 596).

Dearming service times are set by USERF(75). The service times are set by draws from triangular distributions which are chosen based upon the weapon status which was set while the aircraft was on its mission. The dearming times are user definable in Subroutine INTLC (Appendix B, pages 143 and 144). Aircraft with armament malfunctions take longer to dearm than good aircraft. After dearming is completed, good aircraft acquire a parking space and proceed to their squadron area.

Hot Pit Refueling

Aircraft enroute to the squadron area from dearming will Hot Pit refuel if their taxi route makes it convenient, and if they are not being parked in a shelter. Aircraft in shelters are always refueled in the protection of the shelter. In order to Hot Pit refuel, the aircraft must acquire a Hot Pit resource and fuel must be available. If no fuel is available, the aircraft will bypass the Hot Pit area. POL used in refueling is decremented from the overall base supply. Aircraft which break are taxied or towed to appropriate destinations. Towed aircraft have their pilots separated (PSEP). Refer to the graphical network for complete routing possibilities (Appendix A, page 89).

Engine Shutdown and Reconfiguration

When an aircraft reaches the squadron area, it is parked in its assigned parking space and the engine is shut down. The pilot deplanes (PSEP) and returns to the appropriate pilot ready pool after a short delay for a quick conceptual walk around. If a maintenance failure has occurred, the aircraft is scheduled for repair. Otherwise, the aircraft acquires a crew chief and is checked to see if reconfiguration is required. Often external fuel tanks will have to be hung. Reconfiguration is accomplished by USERF(73) (Appendix B, page 175 to 176). The code is intentionally written to avoid downloading tanks at all costs to avoid having to defuel external tanks. When reconfiguration has been completed, or if it is not required, the aircraft enters turnaround service. If the aircraft Hot Pit refueled on its way to the squadron and then had tanks hung on reconfiguration, the fuel which will be required in turnaround service is recomputed. Refer to the graphical network for complete routing possibilities (Appendix A, page 92).

Turnaround Service

The aircraft are turned in a parallel, or concurrent service operation. The mechanics of concurrent service were covered earlier in this chapter. Services performed are maintenance post-flight (and preflight), rearming, and refueling. Rearming requires the acquisition of a rearming crew resource. Rearming service time is set by USERF(76) (Appendix B, page 178). If no armament is required, the aircraft is routed around arming. Refueling is accomplished in shelters by pipelines for aircraft parked in shelters. Shelter refueling service time is set by USERF(78) (Appendix B, page 179). Aircraft in non-sheltered parking spaces are refueled by trucks. Refueling rate and rearming times are user specified in Subroutine INTLC (Appendix B, pages 143 and 144). If fuel is not available, aircraft wait in their parking spaces until fuel is available.

Maintenance

Overview

Aircraft may go into the maintenance function of the model from nearly any point in their flow on the network. Aircraft break when the total engine running time exceeds the NTOF of an aircraft system. The total engine running time is updated after each activity where the aircraft engine is running. Each time the total engine running time is updated, a check is made to see if any failures have occurred. If any failures have occurred, the maintenance failure code is updated. At each place on the network where the failure code is updated, routes have been included for the aircraft to go to maintenance.

Maintenance Control

When aircraft enter the maintenance complex, they are initially divided into three groups. One group has only minor failures (failure level less than four). Another group is made up of aircraft with major failures (failure level equal to or greater than four). Aircraft with any battle damage are grouped together regardless of their maintenance failure status.

Aircraft in the minor problem group are sent to squadron level maintenance for service. Aircraft with battle damage are examined to see if they are repairable. If they are judged repairable, the battle damage code is converted to a maintenance failure code using user provided codes in data statement NBATREP (Appendix B, page 142, line 544). Repairable aircraft are routed to the appropriate maintenance function. Aircraft which will be scrapped have Attribute(18) set to 999999, and they are routed out of the system to a file (JUNK). Ordnance is downloaded when an aircraft requires service at wing or by an MMT unit.

Aircraft with major problems arrivc at node MCON where routing occurs based on the following decision rules:

(1) Repair aircraft at wing if a required shop is free,

(2) Repair with an MMT unit, if a required MMT unit is free,

(3) Wait for repair at wing, if waiting space is available,

(4) Go to squadron maintenance and repair any minor problems, and then wait for an MMT unit.

The wing shops, squadron shops, and MMT unit system capabilities are presented in Figure 3.3, along with the failure levels which can occur. Refer to the structural model for a graphical depiction (Appendix A, pages 101 and 102).

Priority Processing Code

Aircraft are processed using a priority code. The object of the code is to have the least broken aircraft repaired first and then returned to flying operations. For aircraft entering wing or MMT service, the failure levels of all systems with level four and five problems are added together to yield a value. This is done by USERF(38) (Appendix B, page 162). The value is placed in Attribute(17) and aircraft are processed using a low value first priority (Appendix A, page 11).

A similar process is used for aircraft going to squadron maintenance. The code is calculated by USERF(39), which adds the failure levels of all systems with failures greater than or equal to two (Appendix B, page 163). Aircraft are processed low value first (LVF) based on the code (Appendix A, page 11).

Wing Maintenance

Aircraft arriving to wing are assigned their priority

	Shop Number		
Aircraft System	Wing	MMT	Sqdn.
Electrical	3	1	3
Engine/Fuel	2	2	2
Hydraulic/Pneumatic	1	3	1
Airframe (struts/tires)	1	4	1
Comm/Nav/Instruments/Radar	3	5	3
Fire Control/Wpns Release	4	6	4

Ievel	Definition	Flyable
0	no discrepancy	yes
1	flyable discrepancy	yes
2	minor, short repair time	no
3	minor, long repair time	no
4	major, short repair time	no
5	major, long repair time	no

Fig. 3.3 Maintenance Unit Capabilities and Failure Ievels

code, and if a required shop is open, they enter service. Otherwise, the aircraft go to a waiting pool (WGPOOL) where they await service. When an aircraft completes service, it frees the shop resource, resets its failure code and NTOF for repaired systems, and opens the gate for the waiting pool (WGPOOL) so that waiting aircraft can seek service. The gate for the waiting pool at MMT is also opened. If no aircraft at wing takes the free shop, an aircraft awaiting MMT service is towed to wing if it requires that wing shop service. Aircraft in both waiting pools loop back to their respective pool if they are unable to use the shop which was freed.

If the aircraft which freed the wing shop has been completely repaired, it proceeds to turnaround service. If the aircraft still has a major problem, it tries to get into the required wing shop. If service is unavailable, it goes to the waiting pool.

During wing service all minor problems on an aircraft are assumed to have been repaired along with the major problems. No delays are added for this service. The failure codes and NTOFs are merely reset. The services provided by each shop were described earlier in Figure 3.3. Refer to the structural model for a graphical presentation (Appendix A, pages 106 and 107).

MMT Maintenance

MMT units are mobile maintenance teams of wing specialists which travel to the squadron parking areas to do major repair work. Aircraft requiring an MMT are assigned a priority processing code as at Wing Maintenance. There are MMT units which are equipped to work on each aircraft system, and only that system. The systems were previously presented in Figure 3.3. If a required unit is available, the aircraft enters service. Otherwise, it goes to squadron maintenance if it has minor problems which require repairs. When service is completed at squadron, or if no minor problems require repair, the aircraft goes to a waiting pool (MMTPOOL) and awaits an MMT unit resource.

When an aircraft completes service, it frees the MMT unit, resets the appropriate failure code digit and NTOF, and opens the MMTPOOL gate. Opening the gate allows the waiting aircraft to seek service from the MMT unit which was freed. The aircraft in the pool with the highest priority (lowest value in Attribute(17)) which requires the free MMT unit enters service and the other aircraft loop back to the waiting pool (MMTPOOL).

Aircraft in the MMT waiting pool (MMTPOOL) are allowed to seek service at wing, after aircraft waiting at wing (WGPOOL). This insures that the Wing Maintenance facility is effectively utilized. If no aircraft in the wing waiting pool (WGPOOL) require a free wing shop, an aircraft awaiting an MMT unit and requiring the free wing shop will be towed from the squadron area to Wing Maintenance.

If an aircraft freeing an MMT unit has completed all service, it proceeds to turnaround servicing. If the aircraft requires further service, it tries to obtain an appropriate MMT unit for major repairs, or it goes into squadron maintenance to complete minor repairs. Unlike wing shops, MMT units do not perform concurrent repairs of minor problems. Refer to the structural model for a graphical presentation (Appendix A, pages 112 and 113).

Squadron Maintenance

Each squadron has a maintenance facility with shops which are functionally the same as Wing Maintenance. Functions were previously shown in Figure 3.3. Aircraft minor problems are repaired concurrently based on the aircraft Priority Processing Code. Concurrent servicing was covered earlier in this chapter. When all servicing is completed, failure codes and NTOFs are reset. Aircraft which have major problems remaining go to MMT units (DLMT then MMT) for maintenance service. Aircraft which have completed maintenance go to turnaround servicing. Refer to the structural model for a graphical presentation (Appendix A, pages 121 and 122).

Each squadron has its own facility in the network. However, when repairs are completed, the aircraft are matched back together (SQMA) in a common network based on the aircraft's unique tail number. After the aircraft has been reassembled and failure codes and NTOFs are reset, the aircraft is routed to turnaround servicing.

Executive Network

Overview

The Executive Network controls the airfield model. The user must input the times to initiate the key activities in Subroutine INTLC (Appendix B, pages 144 and 145). The key activities use discrete event orientation and are initiated by Event 1, the Master Clock, based on the user specified times. As shown in Figure 3.4, the key activities are Scheduler (Events 2 and 3), Night Parking (Events 4 and 5), QRA Changeover (Event 6), Resupply, and Reconfigure (Event 7). Refer to the structural model for a graphical presentation (Appendix A, pages 126 and 127).

Scheduler

The scheduling routine operates during the day. The first step is the initialization of the user specified frag. requirement for that day (Event 2). Specification of the fragmentary order is covered in Appendix C. When initialization is complete, the Scheduler (Event 3) uses Subroutine ORGANPT (organize pilots) to form and initiate the launch of flights according to the frag specifications the user has made in Subroutine USERI (Appendix B, page 171). The decision rules used to organize a flight (ORGANPT) are in Appendix B, pages 208 and 209.

Night Parking

The aircraft assigned to the airfield are day only, clear weather, fighter-bombers. The first and last times on

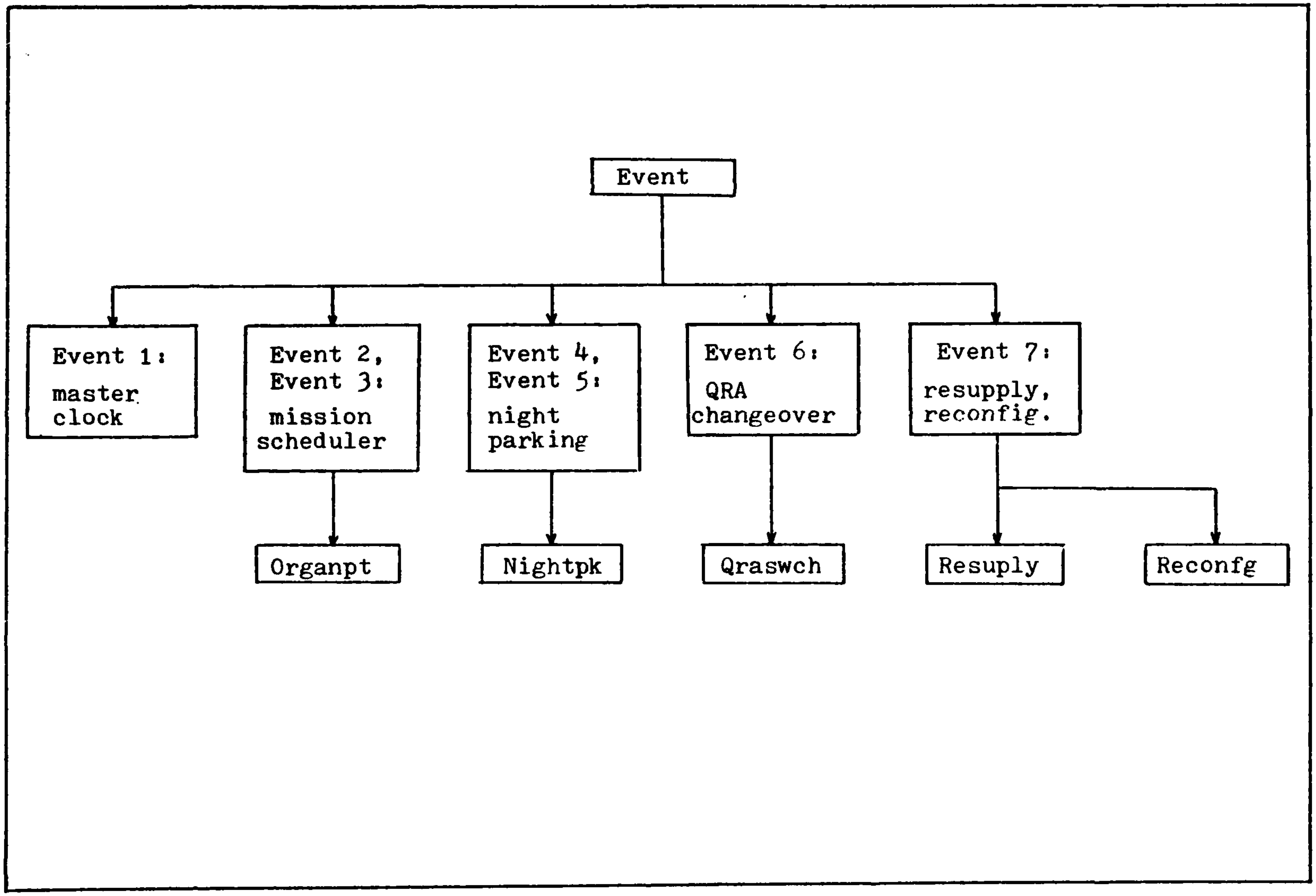

Event
Event 1: master clock
Event 2, Event 3: mission scheduler
Event 4, Event 5: night parking
Event 6: QRA changeover
Event 7: resupply, reconfig.
Organpt
Nightpk
Qraswch
Resuply
Reconfg

Fig. 3.4 Executive Network Routines

target (TOT) can only be scheduled between first light and civil twilight. The aircraft may operate in periods of darkness enroute to and from targets, when under radar coverage of friendly radars. The only conceptual activities which occur during the night are the repair, reconfiguration, and ground movement of aircraft. To gain the greatest protection from aerial attacks, the aircraft are doubled up in shelters during the night, except for those shelters housing a single QRA aircraft.

Event 4 conducts the initial parking, and Event 5 provides for ongoing parking of aircraft which are completing turnaround servicing. This insures that aircraft will be parked in the safest available space.

QRA Changeover

Event 6 initiates Subroutine QRASWCH, which changes over the pilots on QRA during the evening (Appendix B, pages 214 and 215). If a squadron has been disbanded and replaced by a replacement squadron during the day, the aircraft are also changed over.

Resupply and Reconfiguration

Both Subroutines RESUPLY and RECONFG are initiated by Event 7. RESUPLY determines if a squadron requires a replacement unit to be flown in at approximately midday the following day. The determination is made by comparing the current number of operational aircraft in a squadron to a user specified lower limit LIMITAC. LIMITAC is specified in Subroutine USERI

(Appendix B, page 133).

If a squadron requires replacement RESUPLY (Appendix B, pages 216 to 220) farms out the aircraft and pilots to the sister squadrons in the same wing. The decision rules for distributing the aircraft and pilots are in Appendix B, pages 217 and 218. The QRA aircraft and pilots of the squadron remain on alert until QRA Changeover in the evening of the day the replacement squadron arrives.

Subroutine RECONFG (Appendix B, pages 221 and 222) reconfigures the aircraft during the night. The ratio of configurations by area is specified by the user in Data Statement INITAC in Subroutine USERI (Appendix B, page 131). Refer to Appendix C for complete instructions on how to set up the values of INITAC. The routine determines how many squadrons will be configured for each geographic area the wing will be flying to on the next day, and then changes the configurations of the aircraft. It is assumed that there is sufficient time during the night to reconfigure all the aircraft and so service times are not accounted for.

Event 7 (Appendix B, pages 204 and 205) also causes a printout of the contents of the file (JUNK) holding aircraft which have left the system during the day. The contents of the file are then destroyed by opening the JUNK gate and allowing the entities to terminate. See the structural model for a graphical presentation of this operation (Appendix A, page 126).

Summary

Figure 3.5 synthesizes the material covered in this chapter and graphically depicts how the SLAM network model interfaces with both the network oriented and discrete event oriented FORTRAN routines. The reader who desires a more detailed understanding of the model should refer directly to the SLAM and FORTRAN coding. These listings are contained in Appendices A and B, respectively, and they are extensively commented. The structural model graphical depictions in Appendix A provide an excellent means of gaining insights into how the network structure functions. To complete familiarization with the model, refer to Appendix C and Annex A.

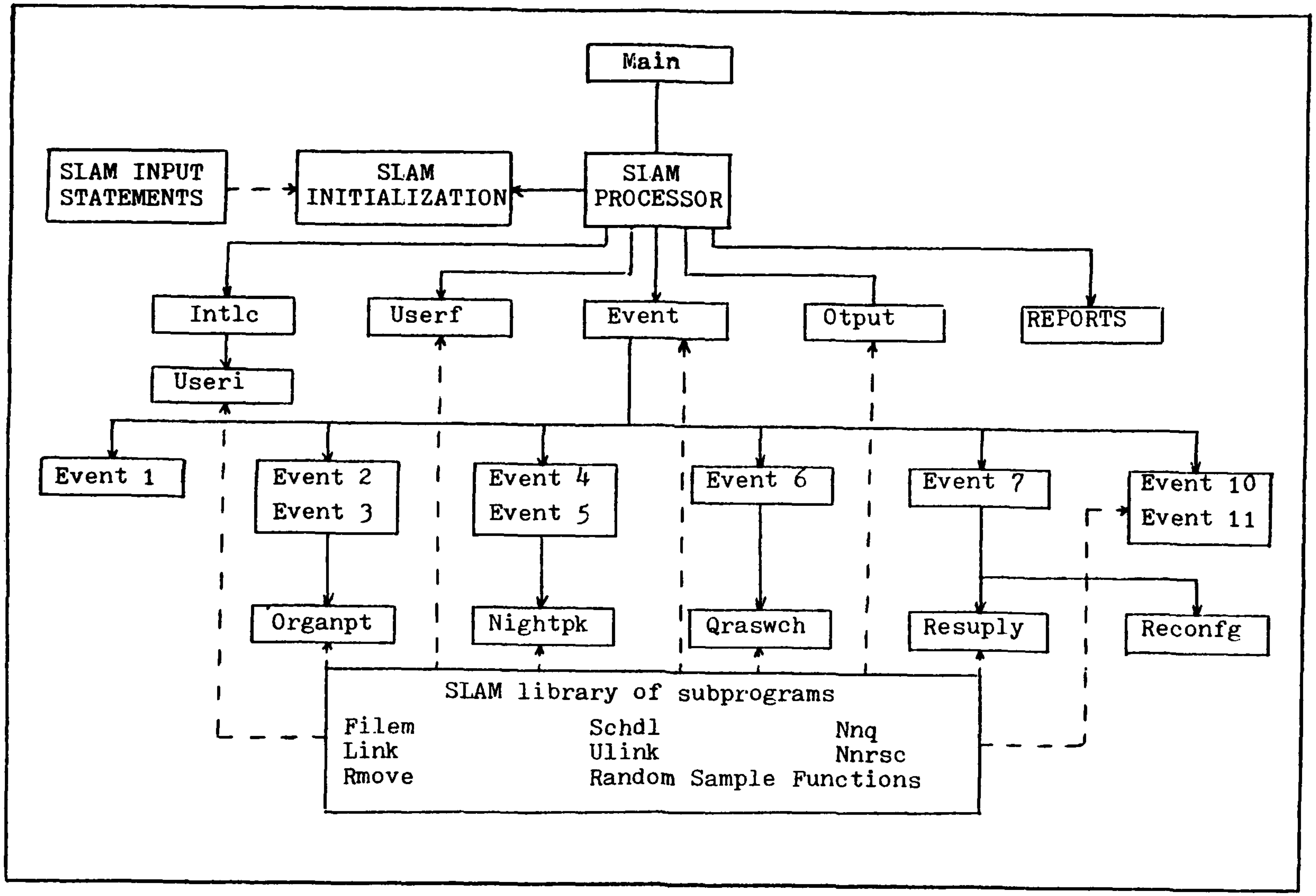


Fig. 3.5 The Airfield Model

IV. VERIFICATION AND VALIDATION

Overview

Prior to beginning construction of the model, several policies, or decision rules, were developed. These policies came to have a pervasive influence on the model, and hence, verification and validation. The policies included using an absolutely explicit network, with activities modeled discretely enough that tight triangular distributions could be used for service times. In addition, the model was not limited by design to only being useful for experimentation on element criticality. The last policy dictated modular construction with careful testing at each stage of assembly. Each of these policies is covered in more detail in the remainder of this section.

The first policy was to put every possible part of the model explicitly on the network, unless that was clearly impractical. This approach was used, not only to allow quicker and easier understanding of the model by a user, but also to make the structural assumptions apparent to aid in verification of the model. As a side benefit, it was hoped that making the model structurally explicit would enhance user acceptance--which may be the only measure of validity available for a model of a system this complex.

Another policy was established that dictated the model would be constructed in a modular fashion--with careful testing of each section--and testing of the whole as sections were added. Using this method of construction, verification was accomplished during all phases of construction. During the Q-GERT modeling project mentioned in Chapter I, it became apparent that a useful airfield model would undoubtedly be fairly large and complex. It was, therefore, imperative that verification be accomplished along with construction, using a carefully designed plan of assembly and testing.

The third policy also resulted from lessons learned in the preliminary Q-GERT project. The Q-GERT airfield model produced wide variance in the response variable, which had also been sorties generated over a three-day period. This was not surprising due to the nature of fighter operations. Variance reduction techniques are available for use in modeling, but they must be built into the model from inception. No single variance reduction technique can be applied to an entire model such as the airfield model. Certain techniques may have some applications for small segments of this model, but they were not attempted due to the limited time available for construction and experimentation with the model. In experiments with the Q-GERT model the variance had been reduced by making a large number of runs, but the size of the SLAM model and its run time would not permit this technique to be used. It was cost prohibitive.

The third policy was designed to address the issue of variance reduction. When faced with a choice of how much detail to include in conceptual airfield operations, the choice was resolved in favor of the option which yielded enough detail so that activities could be defined by very tight triangular distributions. The overall goal of this policy was to attempt to reduce variance, to the extent possible, with activities that could be discretely pinned down. Activities were not modeled merely because they could be modeled, but activities were modeled to the level of detail where activity times could be well defined with a high degree of certainty--approaching a deterministic value.

The fourth policy involved trying to make the model useful to a wide audience. When faced with a choice of whether or not to model activities or elements which were not germane to experimentation on element criticality, the decision was based on the amount of effort required. If a subject for experimentation could be envisioned for the activity or element, and if the addition could be made with only a slightly greater effort, the addition was made. If reconceptualization or structural changes were required, most often the decision was negative. One exception was made when a structural change was made to add the ability to spare aborted aircraft at preflight.

Using the policies described above, the structural model was conceptualized and a plan for construction, assembly, and testing was created. This plan was modified as required when problems were encountered, but in the main, the plan was followed throughout the construction of the model. The plan provided structure to the effort, and the structure enabled verification to be an orderly and planned process at all stages of construction.

The remainder of this chapter is devoted to describing in detail the measures which were taken to insure that the completed model would function as designed, and to the extent possible, emulate the operation of an airfield.

Conceptualization and the Structural Model

During the conceptualization of the structural model, the policies mentioned in the previous section were applied with good effect. Having explicit policies aided in the decision process by helping make very clear what issues had to be considered when making a decision about the model.

When the basic elements which would be included had been identified, the next step was to define the internal operation of an element and the interaction between elements. Each element was dissected to discover how discretely it would have to be modeled. Interactions between elements were studied in detail to insure the process would be modeled faithfully.

Many questions arose regarding SLAM implementation. Each question involving how to implement some conceptual element or process in SLAM became the subject of a test. A very small SLAM model of an element or a process was constructed, coded, and run. Often, several modeling approaches were tested to find out which approach seemed to be the most useful. These tests and their results were invaluable in later construction, and in the modular construction and continual verification process. The tests provided clear examples of how certain sections operated.

Finally, the overall structure of the airfield model was completed. The conceptual loop in which the aircraft would flow was closed. The next step was to computerize the structural model in modular sections.

Computerization and Testing

Modular Construction of the SLAM Network

The first step in computerization was to divide the model into the modular sections called for in the construction plan. Five manageable sections were selected--creation, launch, mission, recovery and turnaround servicing, and maintenance. This is shown in Figure 4.1. The sections are coded individually and then tested by themselves. After a section was individually tested, it was added to the previously constructed sections and the interfaces were tested. The previously added sections were then checked for any required modifications after each addition.

To aid in the process, each section was further divided into subsections which dealt with functionally related activities. For example, launch was subdivided into preflight; start; taxi, marshall, and arm; takeoff; and rejoin. Each of the subsections was coded in SLAM and a supporting FORTRAN routine was developed when required (in some cases, the actual

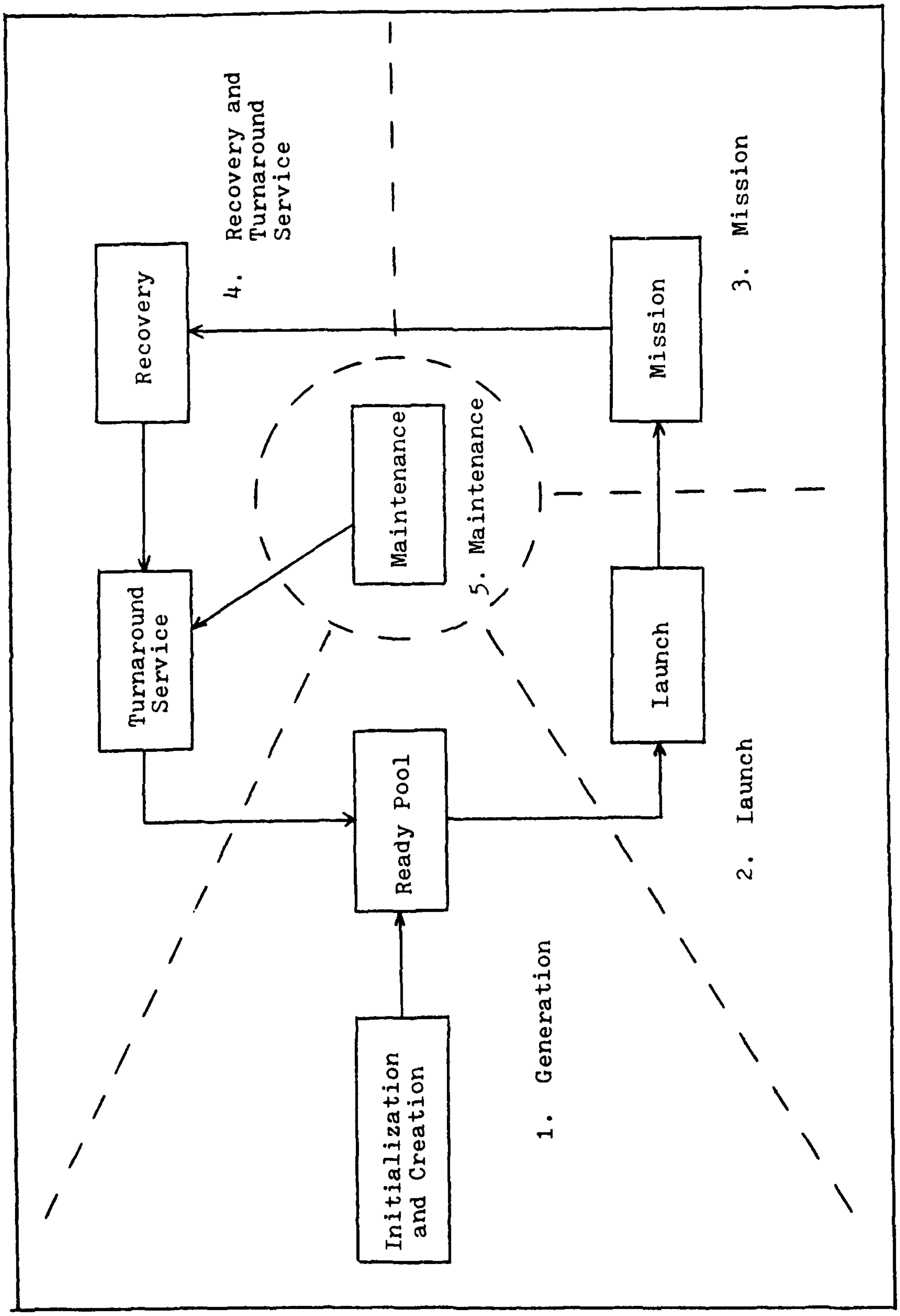


Fig. 4.1 Modular Construction Sections

code which would later be used). A SLAM test front end was then written for the subsection to generate the entities for the test conditions, and a SLAM back end was added to terminate the test and check the results. A test was then run.

Each subsection was individually tested before a section was assembled. In addition, each section was individually tested before the sections were assembled. Each test run utilized the SLAM trace for the entire length of the run. The hard copy of each test run with the trace was carefully examined in detail to verify the proper operation of the subsections and sections.

In assembling the sections, a front to back approach was used with a test run made as each subsection was added. In this manner the same test front end and rear end could be used over and over again. Each addition merely included the next sequential subsection in front of the test rear end. The flow of entities up to the new section was exactly the same as the previous test, which enabled rapid progress in verifying operation up to the new subsection's area. At that point the trace for the new subsection's area was checked for proper operation.

FORTRAN Network Support Routines

Once all the SLAM network sections had been assembled and tested, the network-supporting FORTRAN code was added. Once again, a methodical approach was utilized and each section was treated individually. A section of FORTRAN coding was written to perform a certain required function. If it could be tested all by itself, it was. If it could not stand alone, or if it tested out well, a test of the FORTRAN was run with the SLAM. These tests were run with print statements strategically placed in the FORTRAN code to verify proper functioning of the code. These print statements were supplemented by use of the SLAM trace function.

After all the FORTRAN network support code had been added to each network section, the section was tested individually. When sections were completed, the network structure was assembled and tested section by section, as had been done previously.

Finalizing the Network Model

The network was designed to be driven by an executive network. The testing of the additions of each completed section required that a simple executive network be constructed which would generate a pre-determined number and type of aircraft entities to flow through the system as test cases.

Once the simple executive network was constructed and tested, the major sections of the network were assembled and executed with a trace. Problems which occurred were analyzed using the trace function and FORTRAN print statements. Once problems were identified and solved, the same test was done again with the simple executive network to verify proper functioning of the modified network.

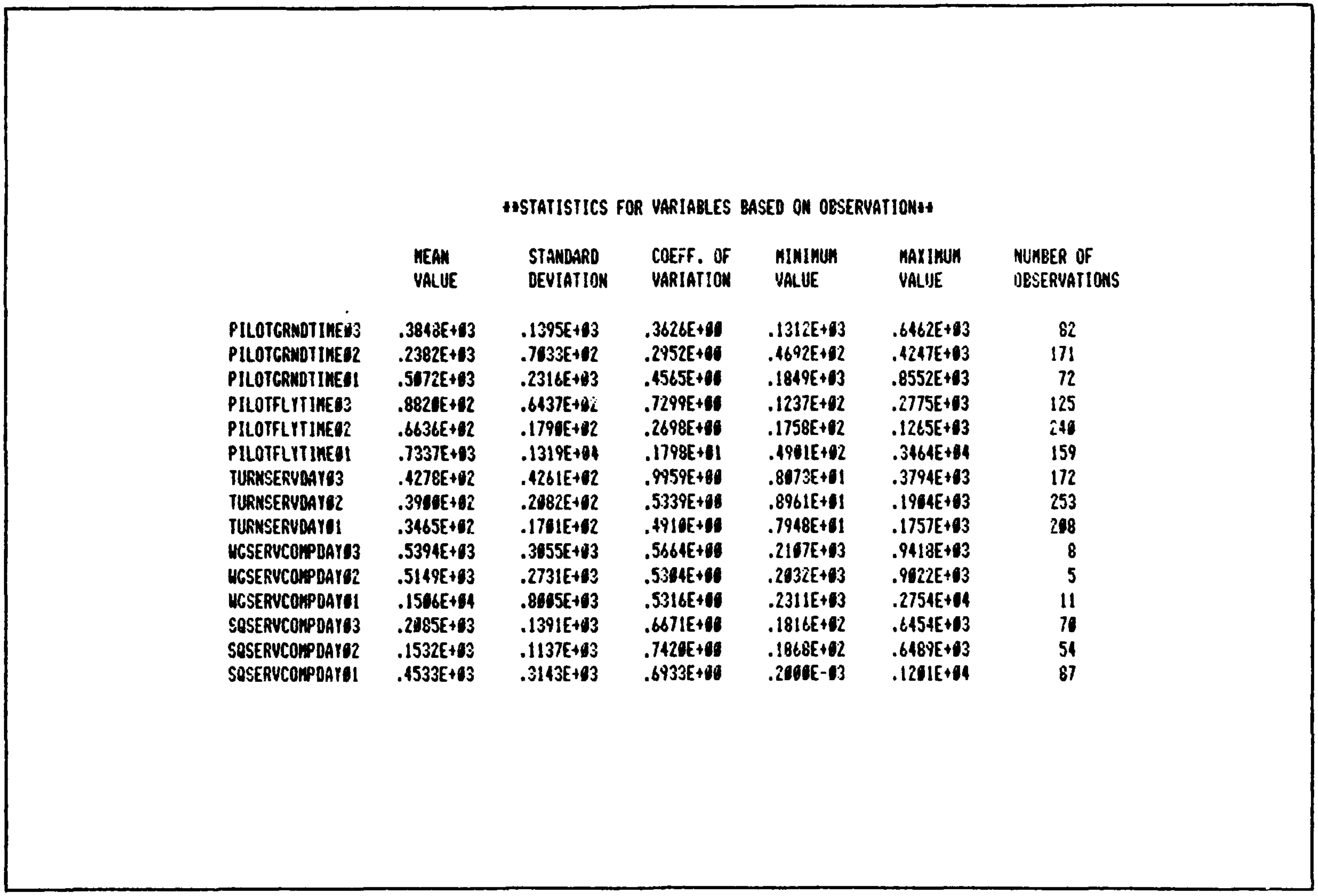
When the entire network had been assembled and successfully tested, the statistical collection nodes (COLCT) were added. The purpose of these nodes was to collect statistics on aircraft and pilot activities to see if the activities were reasonable. The type of information collected is shown in Figure 4.2. After the statistical nodes were added, another series of tests confirmed the continued proper functioning of the network, and the newly added statistical functions.

The Executive Network

With the network functioning properly, it was time to add the discrete event oriented part of the model that would control the major activities during a simulation run. The executive network was designed around a master clock function, which activated major routines (events) at specified times. The routines provide major elements of control for the aircraft and pilot entities in the network. When the executive network was first added, it was supported by some dummy routines. The executive network was tested to verify it functioned properly and then it was added to the model. At that point, the simulation runs ceased being deterministic, as they had been with the simple executive network.

The Discrete Event Oriented Routines

After verifying the proper operation of the executive network, the discrete event oriented FORTRAN routines were added to support the executive network. Each routine was written, tested, and modified as required. When the routine was complete, it was added to the model, replacing the dummy



```
                         **STATISTICS FOR VARIABLES BASED ON OBSERVATION**

                     MEAN        STANDARD     COEFF. OF    MINIMUM      MAXIMUM      NUMBER OF
                     VALUE       DEVIATION    VARIATION    VALUE        VALUE        OBSERVATIONS

PILOTGRNDTIME03     .3848E+03   .1395E+03    .3626E+00    .1312E+03    .6462E+03        82
PILOTGRNDTIME02     .2382E+03   .7033E+02    .2952E+00    .4692E+02    .4247E+03       171
PILOTGRNDTIME01     .5072E+03   .2316E+03    .4565E+00    .1849E+03    .8552E+03        72
PILOTFLYTIME03      .8820E+02   .6437E+02    .7299E+00    .1237E+02    .2775E+03       125
PILOTFLYTIME02      .6636E+02   .1790E+02    .2698E+00    .1758E+02    .1265E+03       240
PILOTFLYTIME01      .7337E+03   .1319E+04    .1798E+01    .4901E+02    .3464E+04       159
TURNSERVDAY03       .4278E+02   .4261E+02    .9959E+00    .8073E+01    .3794E+03       172
TURNSERVDAY02       .3900E+02   .2082E+02    .5339E+00    .8961E+01    .1904E+03       253
TURNSERVDAY01       .3465E+02   .1701E+02    .4910E+00    .7948E+01    .1757E+03       208
WGSERVCOMPDAY03     .5394E+03   .3055E+03    .5664E+00    .2107E+03    .9418E+03         8
WGSERVCOMPDAY02     .5149E+03   .2731E+03    .5304E+00    .2032E+03    .9022E+03         5
WGSERVCOMPDAY01     .1506E+04   .8005E+03    .5316E+00    .2311E+03    .2754E+04        11
SQSERVCOMPDAY03     .2085E+03   .1391E+03    .6671E+00    .1816E+02    .6454E+03        70
SQSERVCOMPDAY02     .1532E+03   .1137E+03    .7420E+00    .1868E+02    .6489E+03        54
SQSERVCOMPDAY01     .4533E+03   .3143E+03    .6933E+00    .2000E-03    .1201E+04        87
```

Fig. 4.2 User Collected Statistics

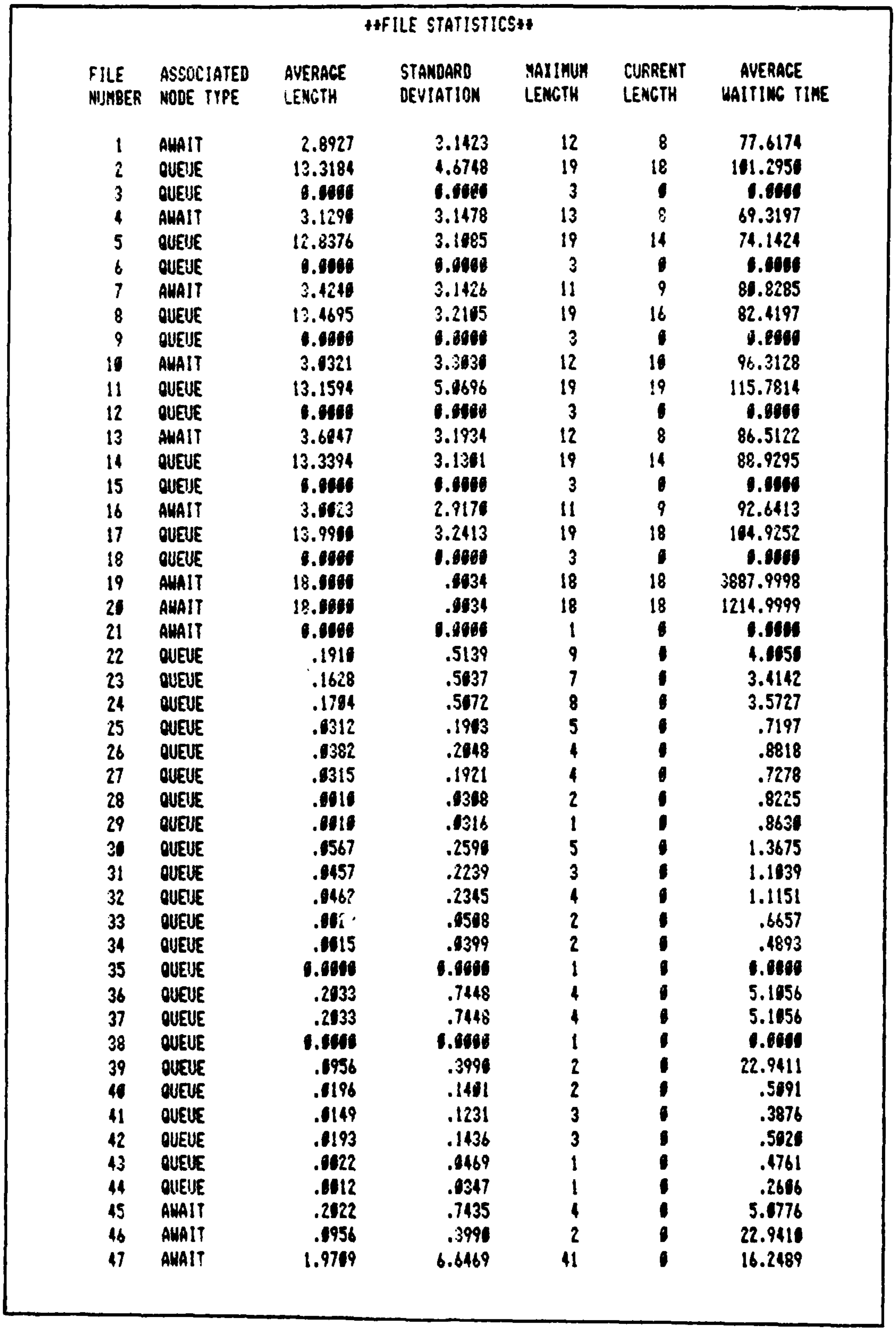
routine initially used. Once again, the SLAM trace and FORTRAN print statements were used to verify the model was functioning properly. Additionally, the SLAM-provided file statistics (Figure 4.3) and the SLAM-provided resource statistics (Figure 4.4) were used to monitor proper functioning of the model.

Troubleshooting Using Statistical Outputs

The file statistics were used to determine if the aircraft entities were being processed as intended, or if any blockages were occurring. The resource statistics were also used to determine if network flow was proceeding properly. As an example, the allocation of the runway resource was originally incorrectly implemented. The problem was only detected after the stress of running two wings of aircraft in the network allowed a rare situation to occur. The problem was detected by observing the current utilization of the runway resource (see Resource Statistics, Figure 4.4) was not zero. Since the model ends simulation at approximately 0400 hours, this was clearly aberrational. A check of the file statistics verified that aircraft entities were jamming up waiting to acquire the runway. Once the problem was identified, it was readily solved by correcting the way the runway resource was allocated.

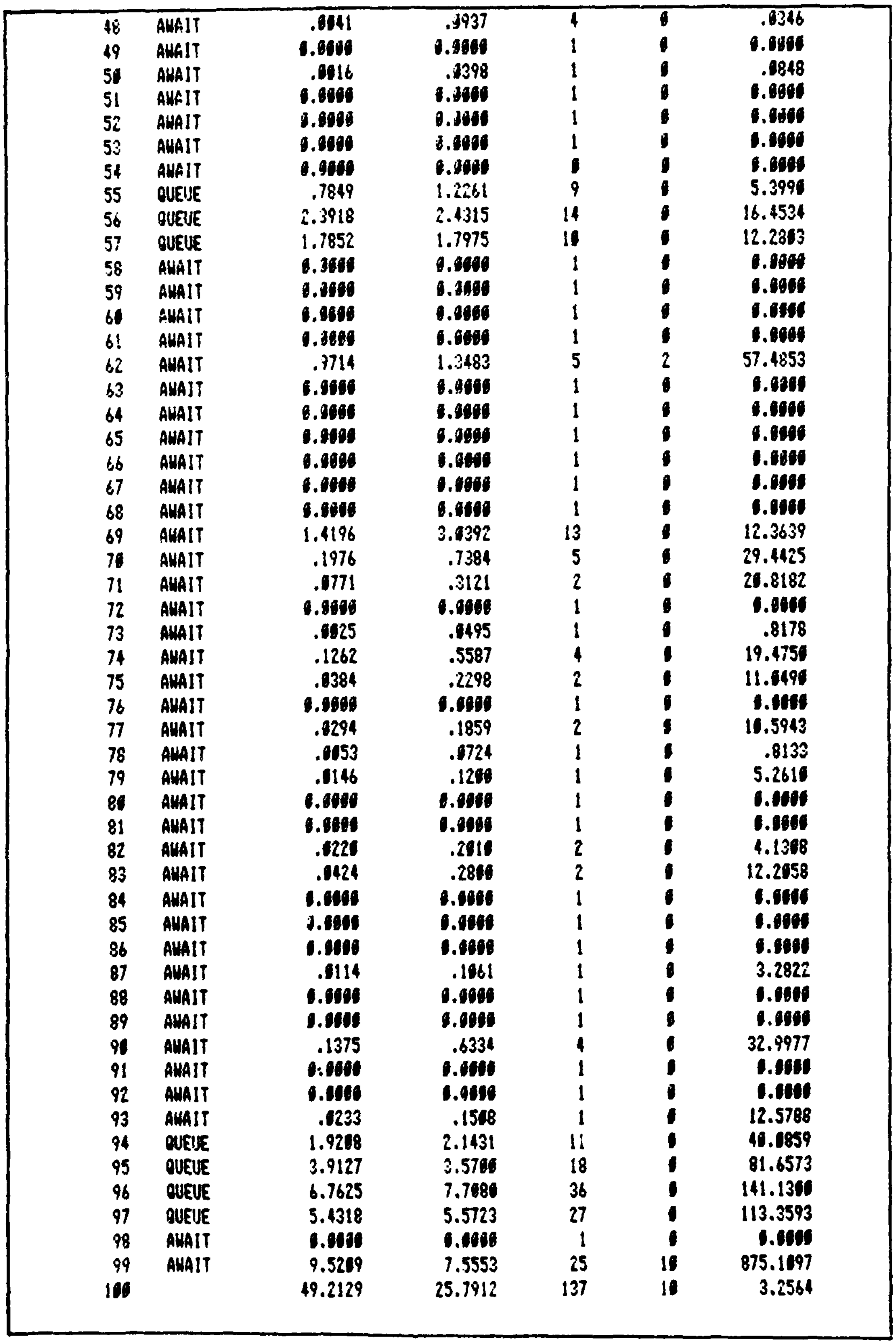
Verification Tests

As a capstone to the previously described efforts at verifying and testing the model as it was constructed, a final



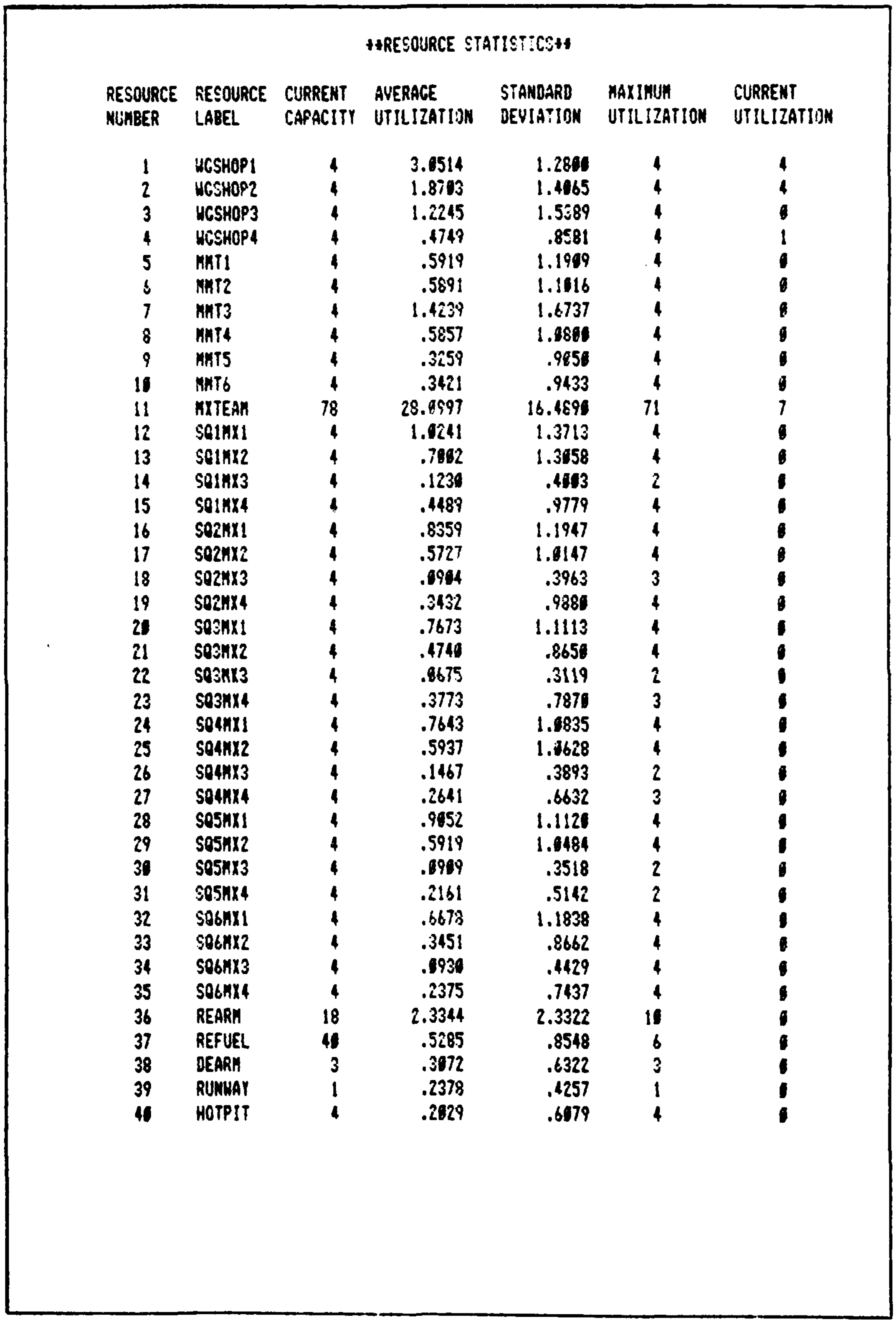
```
                              **FILE STATISTICS**

FILE    ASSOCIATED   AVERAGE       STANDARD      MAXIMUM    CURRENT      AVERAGE
NUMBER  NODE TYPE    LENGTH        DEVIATION     LENGTH     LENGTH     WAITING TIME

  1     AWAIT         2.8927        3.1423         12          8          77.6174
  2     QUEUE        13.3184        4.6748         19         18         101.2950
  3     QUEUE         0.0000        0.0000          3          0           0.0000
  4     AWAIT         3.1290        3.1478         13          8          69.3197
  5     QUEUE        12.8376        3.1085         19         14          74.1424
  6     QUEUE         0.0000        0.0000          3          0           0.0000
  7     AWAIT         3.4240        3.1426         11          9          80.8285
  8     QUEUE        13.4695        3.2105         19         16          82.4197
  9     QUEUE         0.0000        0.0000          3          0           0.0000
 10     AWAIT         3.0321        3.3030         12         10          96.3128
 11     QUEUE        13.1594        5.0696         19         19         115.7814
 12     QUEUE         0.0000        0.0000          3          0           0.0000
 13     AWAIT         3.6047        3.1934         12          8          86.5122
 14     QUEUE        13.3394        3.1301         19         14          88.9295
 15     QUEUE         0.0000        0.0000          3          0           0.0000
 16     AWAIT         3.0023        2.9170         11          9          92.6413
 17     QUEUE        13.9900        3.2413         19         18         104.9252
 18     QUEUE         0.0000        0.0000          3          0           0.0000
 19     AWAIT        18.0000         .0034         18         18        3887.9998
 20     AWAIT        18.0000         .0034         18         18        1214.9999
 21     AWAIT         0.0000        0.0000          1          0           0.0000
 22     QUEUE          .1910         .5139          9          0           4.0050
 23     QUEUE          .1628         .5037          7          0           3.4142
 24     QUEUE          .1704         .5072          8          0           3.5727
 25     QUEUE          .0312         .1903          5          0            .7197
 26     QUEUE          .0382         .2048          4          0            .8818
 27     QUEUE          .0315         .1921          4          0            .7278
 28     QUEUE          .0010         .0308          2          0            .8225
 29     QUEUE          .0010         .0316          1          0            .8630
 30     QUEUE          .0567         .2590          5          0           1.3675
 31     QUEUE          .0457         .2239          3          0           1.1039
 32     QUEUE          .0462         .2345          4          0           1.1151
 33     QUEUE          .001[illegible]         .0508          2          0            .6657
 34     QUEUE          .0015         .0399          2          0            .4893
 35     QUEUE         0.0000        0.0000          1          0           0.0000
 36     QUEUE          .2033         .7448          4          0           5.1056
 37     QUEUE          .2033         .7448          4          0           5.1056
 38     QUEUE         0.0000        0.0000          1          0           0.0000
 39     QUEUE          .0956         .3990          2          0          22.9411
 40     QUEUE          .0196         .1401          2          0            .5091
 41     QUEUE          .0149         .1231          3          0            .3876
 42     QUEUE          .0193         .1436          3          0            .5020
 43     QUEUE          .0022         .0469          1          0            .4761
 44     QUEUE          .0012         .0347          1          0            .2606
 45     AWAIT          .2022         .7435          4          0           5.0776
 46     AWAIT          .0956         .3990          2          0          22.9410
 47     AWAIT         1.9709        6.6469         41          0          16.2489
```

Fig. 4.3.1 SLAM File Statistics



```
 48    AWAIT       .0041      .9937     4    0       .0346
 49    AWAIT      0.0000     0.0000     1    0      0.0000
 50    AWAIT       .0016      .0398     1    0       .0848
 51    AWAIT      0.0000     0.0000     1    0      0.0000
 52    AWAIT      0.0000     0.0000     1    0      0.0000
 53    AWAIT      0.0000     0.0000     1    0      0.0000
 54    AWAIT      0.0000     0.0000     0    0      0.0000
 55    QUEUE       .7849     1.2261     9    0      5.3990
 56    QUEUE      2.3918     2.4315    14    0     16.4534
 57    QUEUE      1.7852     1.7975    10    0     12.2803
 58    AWAIT      0.0000     0.0000     1    0      0.0000
 59    AWAIT      0.0000     0.0000     1    0      0.0000
 60    AWAIT      0.0000     0.0000     1    0      0.0000
 61    AWAIT      0.0000     0.0000     1    0      0.0000
 62    AWAIT       .9714     1.3483     5    2     57.4853
 63    AWAIT      0.0000     0.0000     1    0      0.0000
 64    AWAIT      0.0000     0.0000     1    0      0.0000
 65    AWAIT      0.0000     0.0000     1    0      0.0000
 66    AWAIT      0.0000     0.0000     1    0      0.0000
 67    AWAIT      0.0000     0.0000     1    0      0.0000
 68    AWAIT      0.0000     0.0000     1    0      0.0000
 69    AWAIT      1.4196     3.0392    13    0     12.3639
 70    AWAIT       .1976      .7384     5    0     29.4425
 71    AWAIT       .0771      .3121     2    0     20.8182
 72    AWAIT      0.0000     0.0000     1    0      0.0000
 73    AWAIT       .0025      .0495     1    0       .8178
 74    AWAIT       .1262      .5587     4    0     19.4750
 75    AWAIT       .0384      .2298     2    0     11.0490
 76    AWAIT      0.0000     0.0000     1    0      0.0000
 77    AWAIT       .0294      .1859     2    0     10.5943
 78    AWAIT       .0053      .0724     1    0       .8133
 79    AWAIT       .0146      .1200     1    0      5.2610
 80    AWAIT      0.0000     0.0000     1    0      0.0000
 81    AWAIT      0.0000     0.0000     1    0      0.0000
 82    AWAIT       .0220      .2010     2    0      4.1300
 83    AWAIT       .0424      .2800     2    0     12.2058
 84    AWAIT      0.0000     0.0000     1    0      0.0000
 85    AWAIT      0.0000     0.0000     1    0      0.0000
 86    AWAIT      0.0000     0.0000     1    0      0.0000
 87    AWAIT       .0114      .1061     1    0      3.2822
 88    AWAIT      0.0000     0.0000     1    0      0.0000
 89    AWAIT      0.0000     0.0000     1    0      0.0000
 90    AWAIT       .1375      .6334     4    0     32.9977
 91    AWAIT      0.0000     0.0000     1    0      0.0000
 92    AWAIT      0.0000     0.0000     1    0      0.0000
 93    AWAIT       .0233      .1508     1    0     12.5788
 94    QUEUE      1.9208     2.1431    11    0     40.0859
 95    QUEUE      3.9127     3.5700    18    0     81.6573
 96    QUEUE      6.7625     7.7080    36    0    141.1300
 97    QUEUE      5.4318     5.5723    27    0    113.3593
 98    AWAIT      0.0000     0.0000     1    0      0.0000
 99    AWAIT      9.5209     7.5553    25   10    875.1097
100               49.2129    25.7912   137   10      3.2564
```

Fig. 4.3.2 SLAM File Statistics



```
                              **RESOURCE STATISTICS**

RESOURCE  RESOURCE  CURRENT   AVERAGE      STANDARD   MAXIMUM      CURRENT
NUMBER    LABEL     CAPACITY  UTILIZATION  DEVIATION  UTILIZATION  UTILIZATION

    1     WCSHOP1      4         3.0514     1.2800        4             4
    2     WCSHOP2      4         1.8703     1.4065        4             4
    3     WCSHOP3      4         1.2245     1.5389        4             0
    4     WCSHOP4      4          .4749      .8581        4             1
    5     MMT1         4          .5919     1.1909        4             0
    6     MMT2         4          .5891     1.1016        4             0
    7     MMT3         4         1.4239     1.6737        4             0
    8     MMT4         4          .5857     1.0800        4             0
    9     MMT5         4          .3259      .9050        4             0
   10     MMT6         4          .3421      .9433        4             0
   11     MXTEAM      78        28.0997    16.4890       71             7
   12     SQ1MX1       4         1.0241     1.3713        4             0
   13     SQ1MX2       4          .7002     1.3058        4             0
   14     SQ1MX3       4          .1230      .4003        2             0
   15     SQ1MX4       4          .4489      .9779        4             0
   16     SQ2MX1       4          .8359     1.1947        4             0
   17     SQ2MX2       4          .5727     1.0147        4             0
   18     SQ2MX3       4          .0904      .3963        3             0
   19     SQ2MX4       4          .3432      .9880        4             0
   20     SQ3MX1       4          .7673     1.1113        4             0
   21     SQ3MX2       4          .4740      .8650        4             0
   22     SQ3MX3       4          .0675      .3119        2             0
   23     SQ3MX4       4          .3773      .7870        3             0
   24     SQ4MX1       4          .7643     1.0835        4             0
   25     SQ4MX2       4          .5937     1.0628        4             0
   26     SQ4MX3       4          .1467      .3893        2             0
   27     SQ4MX4       4          .2641      .6632        3             0
   28     SQ5MX1       4          .9052     1.1120        4             0
   29     SQ5MX2       4          .5919     1.0484        4             0
   30     SQ5MX3       4          .0909      .3518        2             0
   31     SQ5MX4       4          .2161      .5142        2             0
   32     SQ6MX1       4          .6678     1.1838        4             0
   33     SQ6MX2       4          .3451      .8662        4             0
   34     SQ6MX3       4          .0930      .4429        4             0
   35     SQ6MX4       4          .2375      .7437        4             0
   36     REARM       18         2.3344     2.3322       10             0
   37     REFUEL      40          .5285      .8548        6             0
   38     DEARM        3          .3072      .6322        3             0
   39     RUNWAY       1          .2378      .4257        1             0
   40     HOTPIT       4          .2029      .6079        4             0
```

Fig. 4.4 SLAM Resource Statistics

series of tests were conducted. A trace was run on the entire model over three days. Examination of the SLAM trace and FORTRAN print statements showed the model to be functioning as intended.

Validation

Overview

There is no clear line of demarkation between what is called verification and what is referred to as validation. The perspective used in this section is that verification efforts involve insuring that the coding and structure of the model produce a simulation which functions as intended by the authors. At the other end of the continuum of the verification and validation process, a modeler would validate a particular model by proving that the model was a true representation of the real system being modeled. This is obviously not feasible for the airfield model. Certain methods were used, however, to insure the model was useful for experimentation, and to approach the area of validation as nearly as can be done with a model of this type. These approaches to validation are covered below.

Scenario Adjustment

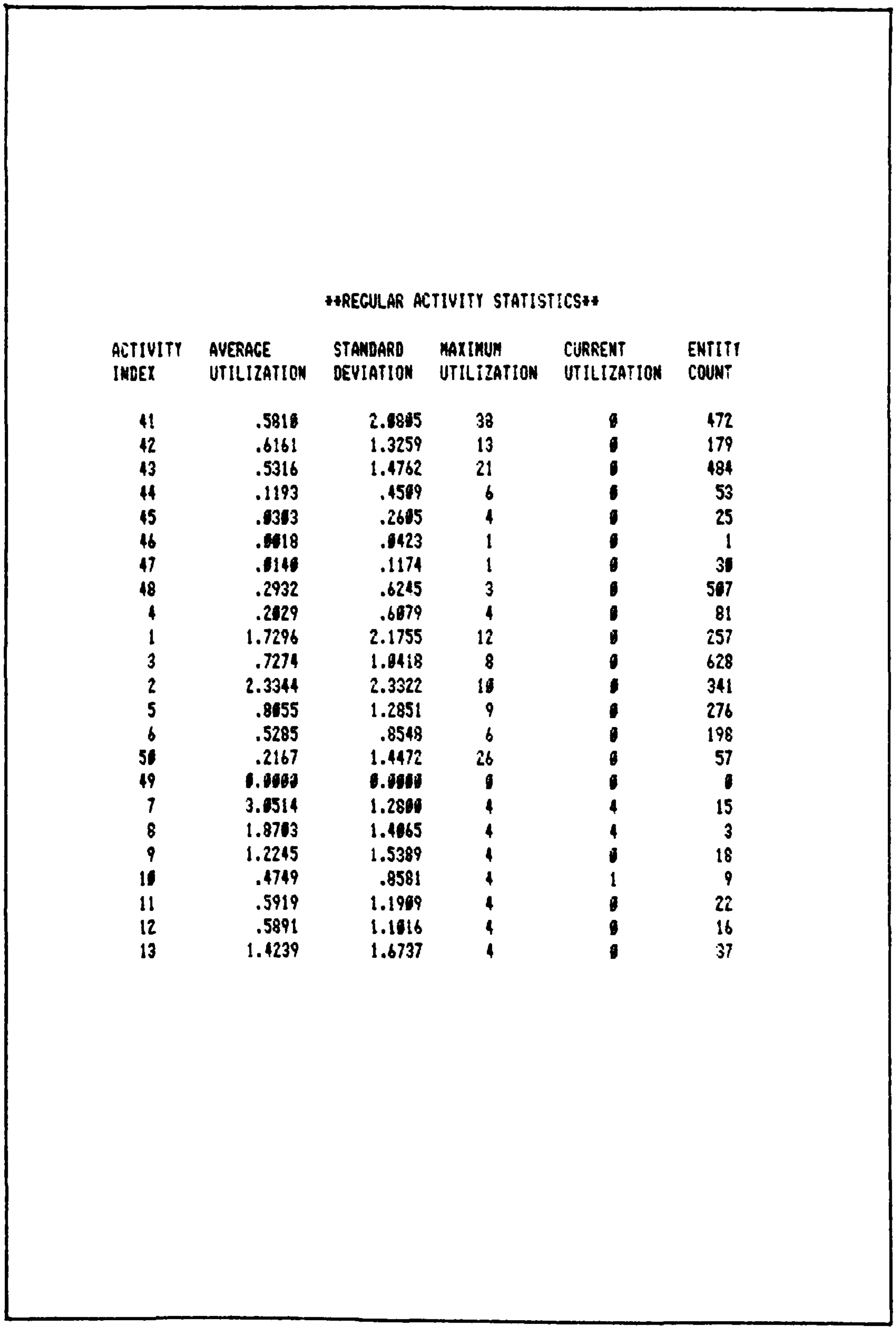
The various parameters of the model were set to values which would be reasonable (mid-range) for a real system. Simulation runs were made and the statistics were reviewed to see if the results were reasonable and believable. Statistics available included the previously mentioned file, resource, and user generated statistics, as well as the SLAM-generated regular activity statistics output (Figure 4.5).

All the statistics generated were in the reasonable range. No aberrational results occurred. The various elements of the airfield appeared to be operating as intended and within the range of activity levels which would be expected.

Statistical Reasonableness

Several attributes had been allocated at the beginning of model construction to serve as statistical collectors. For example, aircraft and pilot Attributes (4), (5), and (6) were designated to store the number of sorties an aircraft or pilot flew on Day 1, 2, and 3, respectively. Refer to Appendix A, pages 3 and 4, for definitions of all attributes. Figure 4.6 shows a sample file dump of file number one (Squadron 1 Aircraft Ready Pool). Two aircraft are shown. The first aircraft flew five sorties on Day 2. The second aircraft flew three sorties on Day 1 and three sorties on Day 2. Other information contained in the attributes listed by the file dump includes parking location, total engine run time, engine run time by day, and aircraft systems NTOF. All the results are within reasonable ranges.

Another file dump, generated by the FORTRAN Event 7 coding, causes the contents of the JUNK file to be listed at the end of each day. After the file is listed, the information is destroyed. The JUNK file contains information on aircraft

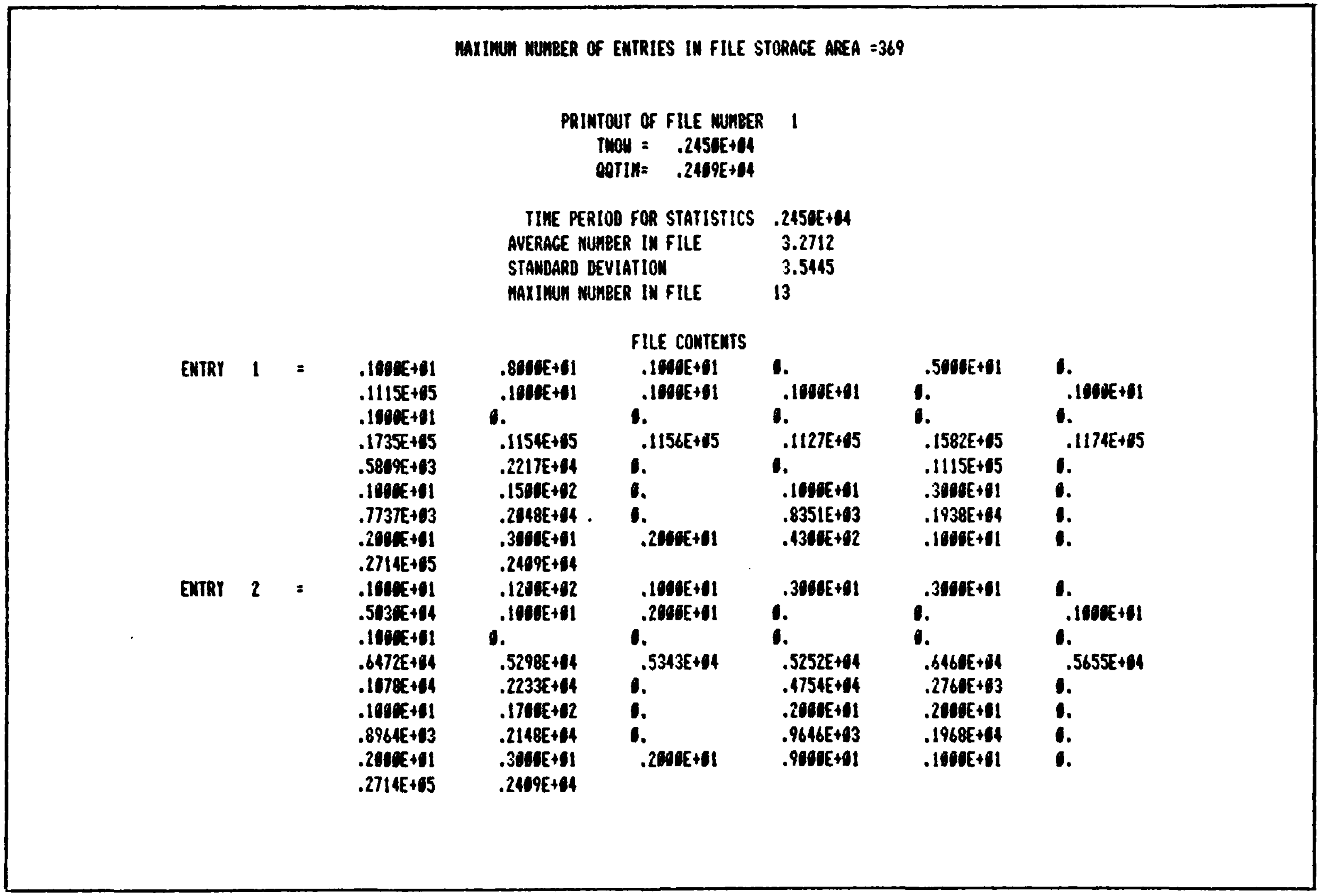


```
                    **REGULAR ACTIVITY STATISTICS**

ACTIVITY  AVERAGE      STANDARD   MAXIMUM      CURRENT      ENTITY
INDEX     UTILIZATION  DEVIATION  UTILIZATION  UTILIZATION  COUNT

   41          .5810     2.0805     38            0           472
   42          .6161     1.3259     13            0           179
   43          .5316     1.4762     21            0           484
   44          .1193      .4509      6            0            53
   45          .0303      .2605      4            0            25
   46          .0018      .0423      1            0             1
   47          .0140      .1174      1            0            30
   48          .2932      .6245      3            0           507
    4          .2029      .6079      4            0            81
    1         1.7296     2.1755     12            0           257
    3          .7274     1.0418      8            0           628
    2         2.3344     2.3322     10            0           341
    5          .8055     1.2851      9            0           276
    6          .5285      .8548      6            0           198
   50          .2167     1.4472     26            0            57
   49         0.0000     0.0000      0            0             0
    7         3.0514     1.2800      4            4            15
    8         1.8703     1.4065      4            4             3
    9         1.2245     1.5389      4            0            18
   10          .4749      .8581      4            1             9
   11          .5919     1.1909      4            0            22
   12          .5891     1.1016      4            0            16
   13         1.4239     1.6737      4            0            37
```

Fig. 4.5.1 SLAM Regular Activity Statistics

```
14        .5857     1.0800     4        0       15
15        .3259      .9050     4        0       19
16        .3421      .9433     4        0       18
17       1.0241     1.3713     4        0       29
18        .7002     1.3058     4        0       16
19        .1230      .4003     2        0        7
20        .4489      .9779     4        0       13
21        .8359     1.1947     4        0       28
22        .5727     1.0147     4        0       15
23        .0904      .3963     3        0        5
24        .3432      .9880     4        0       12
25        .7673     1.1113     4        0       28
26        .4740      .8650     4        0       12
27        .0675      .3119     2        0        4
28        .3773      .7870     3        0       11
29        .7643     1.0835     4        0       23
30        .5937     1.0628     4        0       15
31        .1467      .3893     2        0        9
32        .2641      .6632     3        0        9
33        .9052     1.1120     4        0       30
34        .5919     1.0484     4        0       15
35        .0909      .3518     2        0        6
36        .2161      .5142     2        0        7
37        .6678     1.1838     4        0       18
38        .3451      .8662     4        0        9
39        .0930      .4429     4        0        7
40        .2375      .7437     4        0        8
```

Fig. 4.5.2 SLAM Regular Activity Statistics



```
                     MAXIMUM NUMBER OF ENTRIES IN FILE STORAGE AREA =369


                                PRINTOUT OF FILE NUMBER   1
                                   TNOW =   .2450E+04
                                   QQTIM=   .2409E+04

                              TIME PERIOD FOR STATISTICS  .2450E+04
                             AVERAGE NUMBER IN FILE        3.2712
                             STANDARD DEVIATION            3.5445
                             MAXIMUM NUMBER IN FILE       13

                                         FILE CONTENTS
ENTRY   1    =      .1000E+01   .8000E+01   .1000E+01   0.          .5000E+01   0.
                    .1115E+05   .1000E+01   .1000E+01   .1000E+01   0.          .1000E+01
                    .1000E+01   0.          0.          0.          0.          0.
                    .1735E+05   .1154E+05   .1156E+05   .1127E+05   .1582E+05   .1174E+05
                    .5809E+03   .2217E+04   0.          0.          .1115E+05   0.
                    .1000E+01   .1500E+02   0.          .1000E+01   .3000E+01   0.
                    .7737E+03   .2048E+04 . 0.          .8351E+03   .1938E+04   0.
                    .2000E+01   .3000E+01   .2000E+01   .4300E+02   .1000E+01   0.
                    .2714E+05   .2409E+04
ENTRY   2    =      .1000E+01   .1200E+02   .1000E+01   .3000E+01   .3000E+01   0.
                    .5030E+04   .1000E+01   .2000E+01   0.          0.          .1000E+01
                    .1000E+01   0.          0.          0.          0.          0.
                    .6472E+04   .5298E+04   .5343E+04   .5252E+04   .6460E+04   .5655E+04
                    .1070E+04   .2233E+04   0.          .4754E+04   .2760E+03   0.
                    .1000E+01   .1700E+02   0.          .2000E+01   .2000E+01   0.
                    .8964E+03   .2148E+04   0.          .9646E+03   .1968E+04   0.
                    .2000E+01   .3000E+01   .2000E+01   .9000E+01   .1000E+01   0.
                    .2714E+05   .2409E+04
```

Fig. 4.6 Sample File Dump

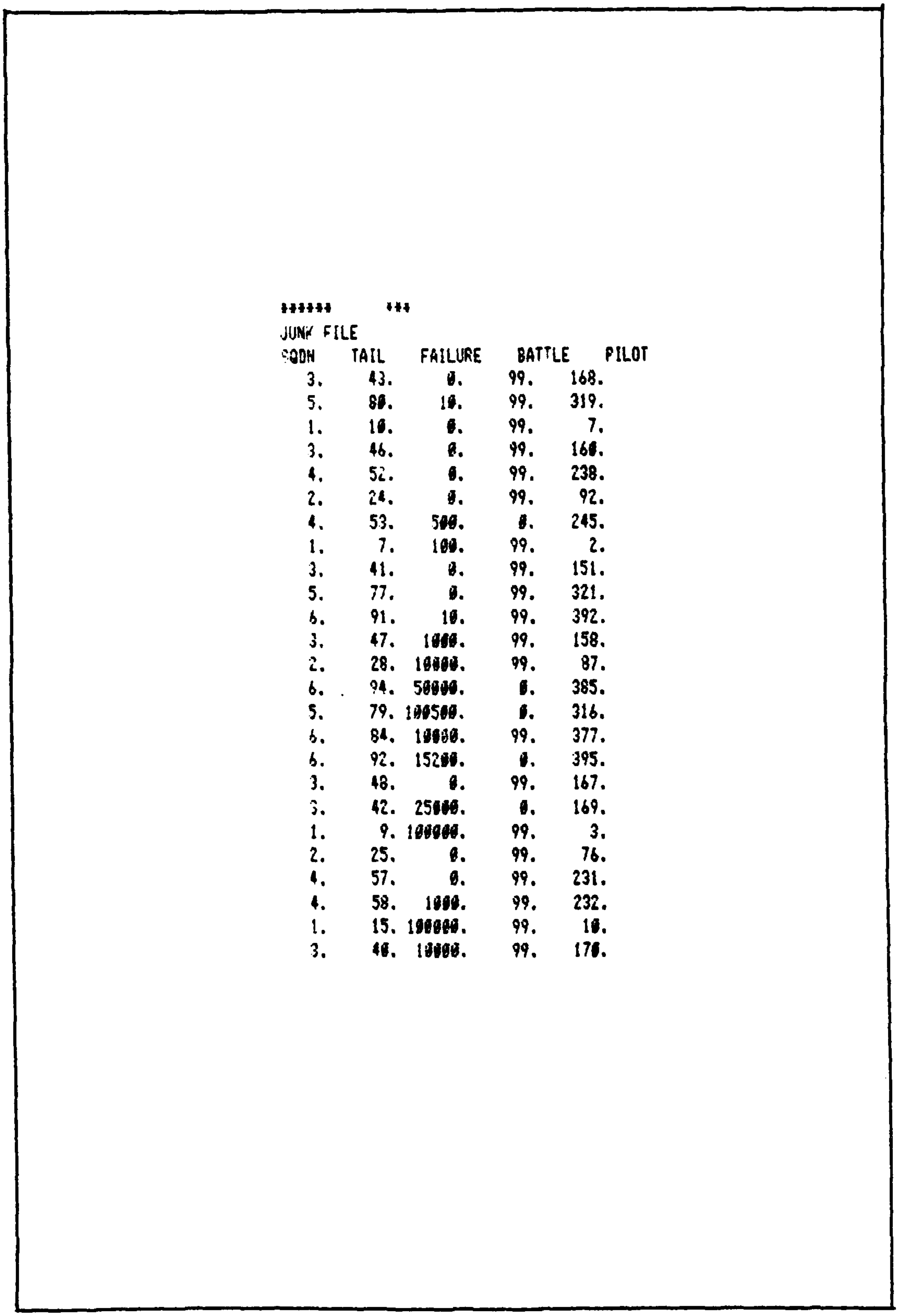
which were eliminated from the system by attrition, maintenance failure resulting in crash, or battle damage resulting in scrapping. The listing shows how many aircraft in each squadron were eliminated, and the reason.

A sample JUNK file dump is in Figure 4.7. The aircraft with a 99 in the BATTLE column were attrited during a mission. Aircraft with a zero in the BATTLE column were lost due to severe maintenance failures which caused a crash. By using the entity count value printed in the regular activity statistics output, and the number of aircraft in the JUNK file, it is possible to cross-check the number of aircraft entities going through each section of the model. This was done on several runs to see if the results were reasonable. They were. The contents of JUNK file dumps have remained within reasonable ranges.

Summary

Verification was a continual process during the construction of the airfield model. Modular construction using the five sections and their functional subsections allowed for ease of testing each section as it was being built-up. The SLAM trace function and FORTRAN print statements combined to give high quality assurance that the coding was functioning properly and as intended.

Validation, as such, cannot be fully accomplished. However, the model is the sum of individual parts which were carefully constructed to emulate the real world of an airfield



```
******      ***
JUNK FILE
SQDN    TAIL    FAILURE    BATTLE    PILOT
   3.     43.        Ø.     99.    168.
   5.     8Ø.       1Ø.     99.    319.
   1.     1Ø.        Ø.     99.      7.
   3.     46.        Ø.     99.    16Ø.
   4.     52.        Ø.     99.    238.
   2.     24.        Ø.     99.     92.
   4.     53.      5ØØ.      Ø.    245.
   1.      7.      1ØØ.     99.      2.
   3.     41.        Ø.     99.    151.
   5.     77.        Ø.     99.    321.
   6.     91.       1Ø.     99.    392.
   3.     47.     1ØØØ.     99.    158.
   2.     28.    1ØØØØ.     99.     87.
   6.     94.    5ØØØØ.      Ø.    385.
   5.     79.   1ØØ5ØØ.      Ø.    316.
   6.     84.    1ØØØØ.     99.    377.
   6.     92.    152ØØ.      Ø.    395.
   3.     48.        Ø.     99.    167.
   3.     42.    25ØØØ.      Ø.    169.
   1.      9.   1ØØØØØ.     99.      3.
   2.     25.        Ø.     99.     76.
   4.     57.        Ø.     99.    231.
   4.     58.     1ØØØ.     99.    232.
   1.     15.   1ØØØØØ.     99.     1Ø.
   3.     4Ø.    1ØØØØ.     99.    17Ø.
```

Fig. 4.7 Sample Junk File Dump

as nearly as can be done. The sum of those carefully built parts should be in reasonable tolerances for reflecting airfield operations. Many statistical measures were built into the model. These measures, in concert with SLAM-generated statistics, have reflected reasonable values across the board. In a left-handed manner they have built confidence in that they have not revealed aberrational results.

Any further validation of the model at this point is impossible. If the model is adopted for use in the future, one more measure of validation will be the degree of user acceptance and confidence the model can garner.

It should be evident at this point that with regard to verification of a complex model, a modular construction approach coupled with careful testing at all stages of construction offers many advantages. With respect to validation, the sum of reasonable parts should yield a reasonable whole. Hopefully, future measures of validity can include user acceptance and confidence.

V. EXPERIMENTATION AND ANALYSIS

Overview

In order to experiment with the airfield model, those factors which could have an influence on the response variable had to be identified. The factors fell naturally into two groups--those internal to the airfield system and those which were external. The important external factors, and two internal factors (rearming crews, preflight delay) became the subjects of sensitivity analysis. The important internal factors became the subjects of factorial experimentation to determine criticality. First, the internal factors were screened using a fractional factorial experiment. The most significant factors identified by the screening experiment were then used as subjects for a full factorial experiment.

Airfield Model Experimental Factors

Internal Factors

The factors included in this grouping have the common characteristics that they are physical entities on real airfields, and they were included in this airfield model. The factors include: dearming crews, rearming crews, shelter refueling, Wing Maintenance shops, MMT units, squadron maintenance shops, and POL supply.

Rearming/Dearming crews were eliminated from further consideration because of the straightforward nature of their activity in terms of training required for qualification. Almost any individual with maintenance experience could be readily cross-trained as an entry level crew member. This is not true for the crew supervisor, but a relatively numerous population would be qualified to fill a supervisory position due to past experience. Such a robust supply of a resource indicates it is not a critical element. Rearming crews were checked for their influence by sensitivity analysis. The model was not very sensitive to changes in the number of rearming crews.

POL supply was eliminated from further consideration. It is obvious that if POL supplies are denied, not one sortie will get airborne. It is equally obvious that total denial of POL is impossible. However, the distribution of POL supplies to aircraft was a matter of interest and, therefore, the elements which distribute POL were examined more closely by experimentation.

External Factors

There were many other factors which could have an influence on the response variable. The factors which could possibly have a significant influence include: MTBF, Beta distribution shape parameters, initially assigned engine run times, attrition rates, geographical area favored by the fragmentary order, and the number of operational aircraft which will trigger the scheduling of a replacement squadron. All these factors are external to the airfield system which generates the sorties. Sensitivity analysis was performed on each of these factors.

Sensitivity Analysis

Each of the important external factors and two internal factors were checked to see how sensitive the response variable would be to changes in the levels of the factors. The levels checked normally included excursions well above, and well below the normal levels of the factors.

The normal levels of the factors were tested in a series of 28 runs to estimate variance. This resulted in a mean value for the response variable of 549.7, with a standard deviation of 37.8. For sensitivity analysis only three runs were performed at each level. This was due to cost considerations.

The results of the sensitivity analysis runs were tested for significance using the Duncan's Multiple Range Test routine contained in the Statistical Package for the Social Sciences (SPSS) computer package (Nie & others, 1982). The null hypothesis of the test is that the means of the results of runs at different levels are equal. The multiple range test divides the means into subgroups such that any two means in a subgroup do not differ significantly (Walpole & Meyers, 1978: 382). The tests were performed at the 95 percent confidence level (ALPHA = .05). Each of the results is described in the remainder of this section.

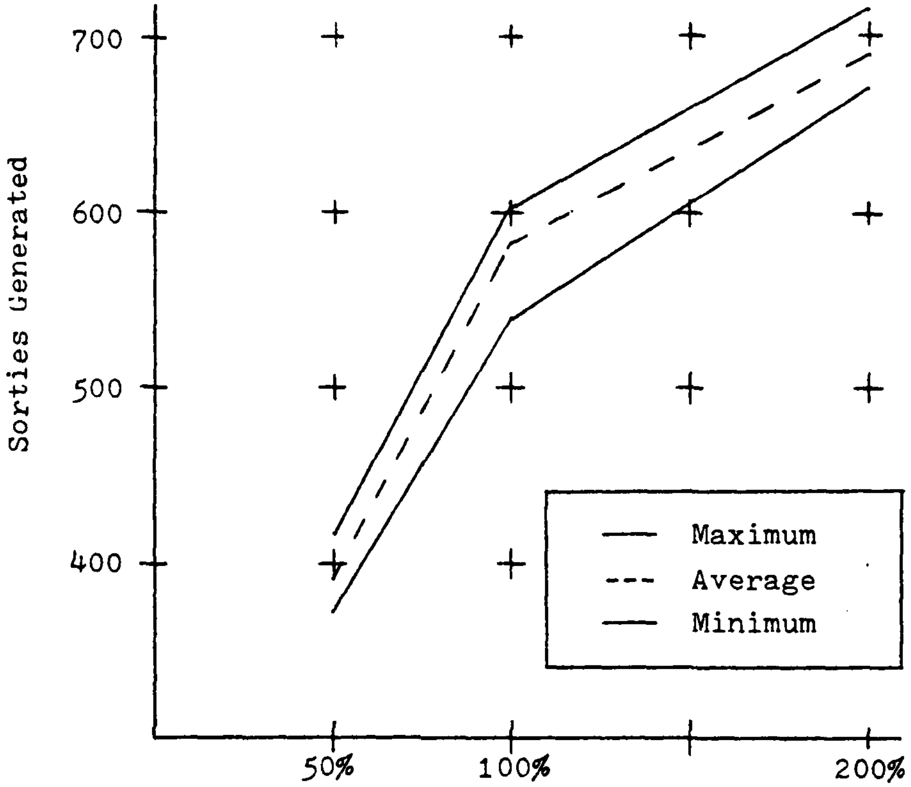
Mean Time Between Failure (MTBF)

The MTBFs of each of the six conceptual aircraft systems was tested at the normal level. For the low value excursion, each MTBF was cut in half. For a high value excursion, each MTBF was doubled. Results are depicted in Figure 5.1. The model is sensitive to changes in MTBF, as expected. The results of the SPSS Duncan's Multiple Range Test are presented as an example in Figure 5.2. Each mean was different at the 95 percent confidence level. Statistically, the changes to the MTBF levels were significant. Users of the model should take care to use the best available estimates of MTBF, and then do careful sensitivity analysis on this particular factor.

Beta Distribution Shape Parameters

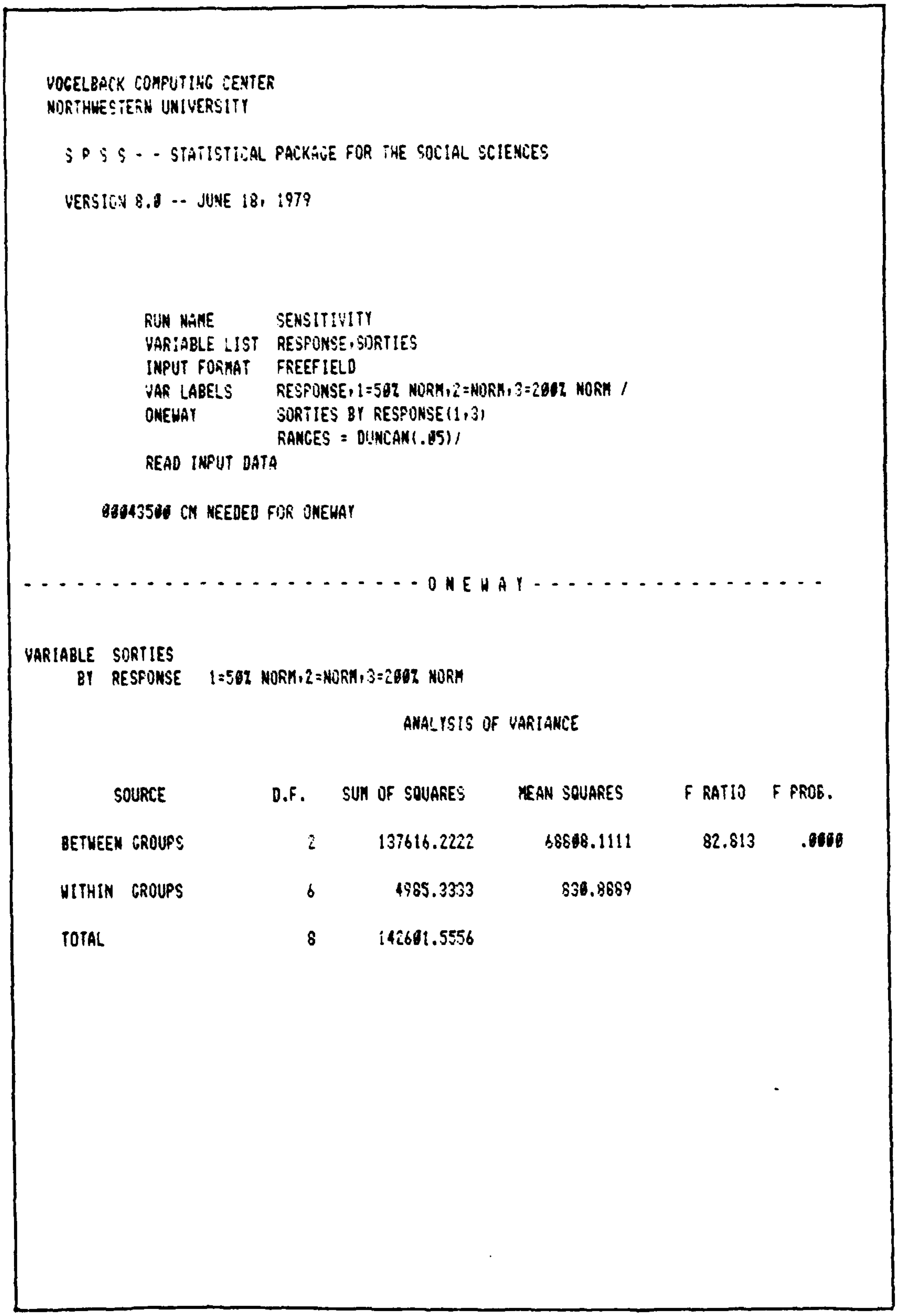
When the shape of the MTBF Beta Distribution is shifted left, the NTOFs are generally of shorter duration. This means systems are less reliable. When the distribution is shaped normally, reliability increases. When the curve is shaped with the shift to the right, the systems are more reliable and more failures occur after longer times of operation.

The sensitivity analysis results are presented in Figure 5.3. There was no statistical difference between the shape shifted left and the normal shape. The shape shifted to the right was statistically different. This was a result of the systems being treated as more reliable. Prior to exercising the model, users should take care to define their own



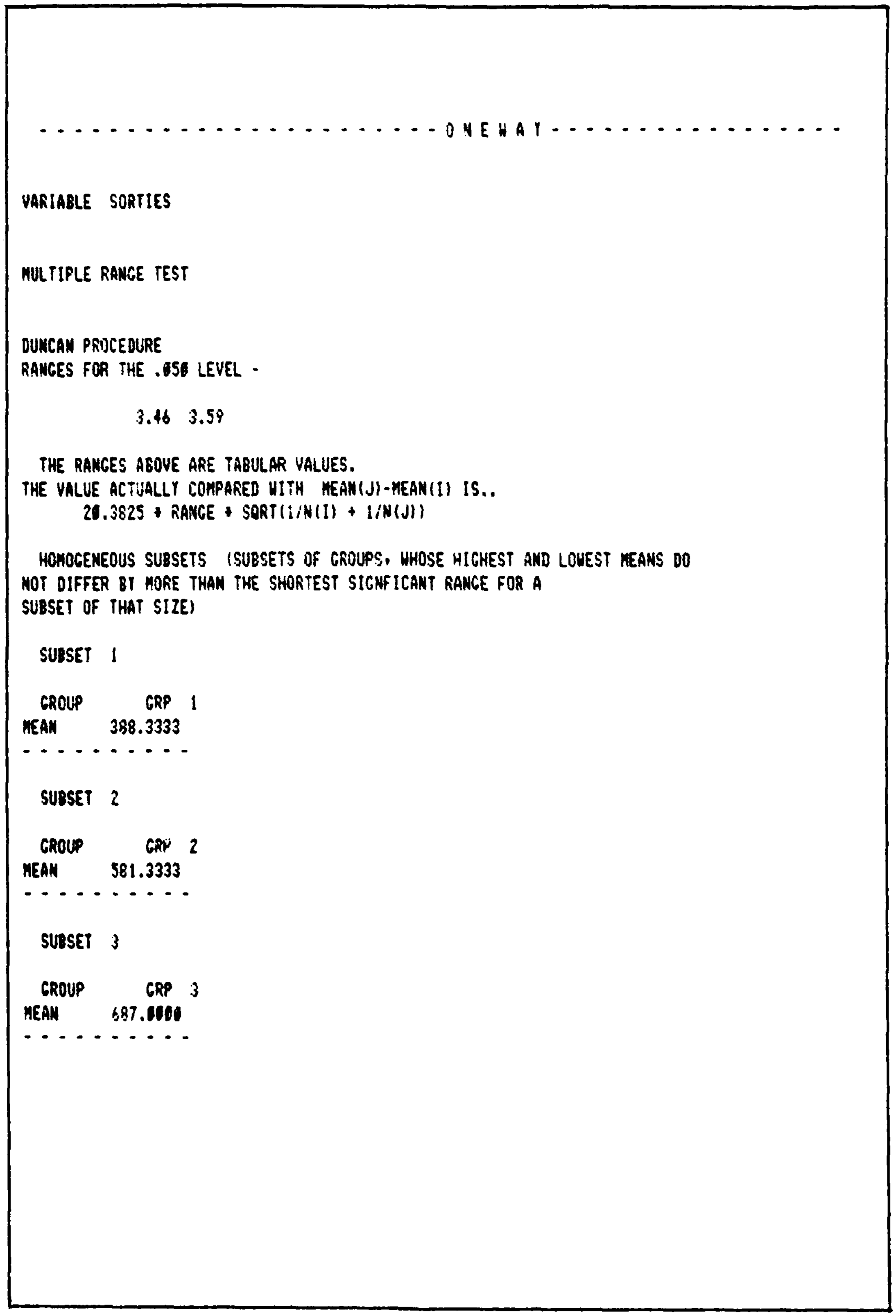
Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
50% of norm	417.0	370.0	378.0	388.3	25.1
100% of norm	539.0	602.0	603.0	581.3	36.7
200% of norm	713.0	677.0	671.0	687.0	22.7

Fig. 5.1 Mean Time Between Failure (MTBF) Sensitivity



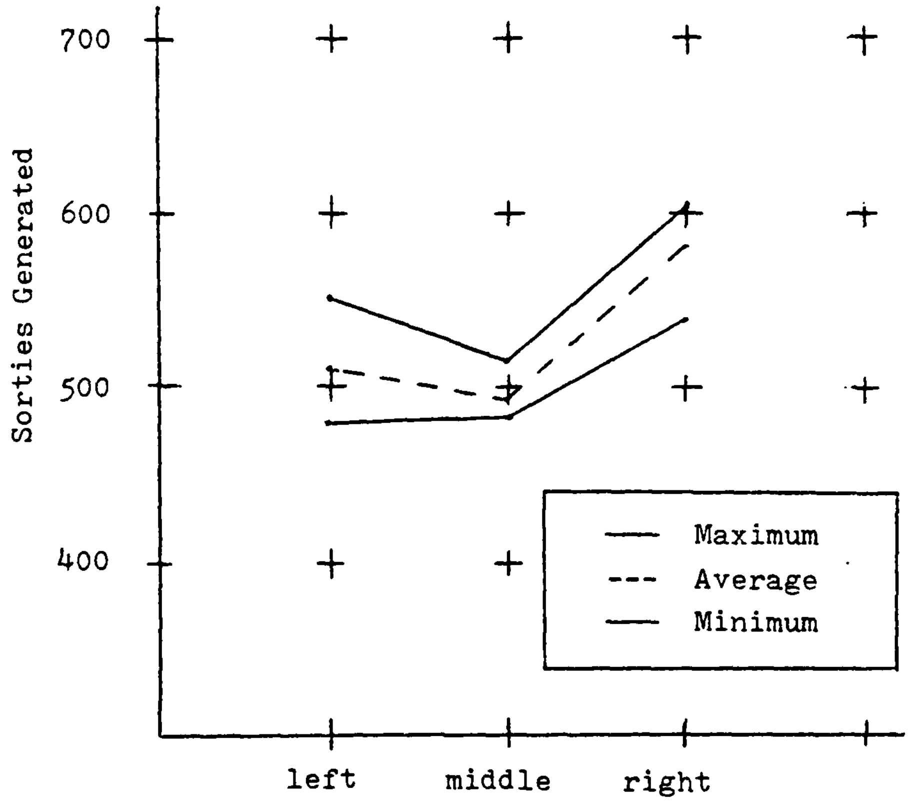
```
VOGELBACK COMPUTING CENTER
NORTHWESTERN UNIVERSITY

  S P S S - - STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES

  VERSION 8.0 -- JUNE 18, 1979



         RUN NAME       SENSITIVITY
         VARIABLE LIST  RESPONSE,SORTIES
         INPUT FORMAT   FREEFIELD
         VAR LABELS     RESPONSE,1=50% NORM,2=NORM,3=200% NORM /
         ONEWAY         SORTIES BY RESPONSE(1,3)
                        RANGES = DUNCAN(.05)/
         READ INPUT DATA

      00043500 CM NEEDED FOR ONEWAY

- - - - - - - - - - - - - - - - - - - - - - - - - O N E W A Y - - - - - - - - - - - - - - - - -


VARIABLE  SORTIES
     BY  RESPONSE   1=50% NORM,2=NORM,3=200% NORM

                                              ANALYSIS OF VARIANCE


          SOURCE            D.F.    SUM OF SQUARES      MEAN SQUARES       F RATIO   F PROB.

   BETWEEN GROUPS              2       137616.2222        68808.1111        82.813     .0000

   WITHIN  GROUPS              6         4985.3333          830.8889

   TOTAL                       8       142601.5556
```

Fig. 5.2.1 Duncan's Test on MTBFs



```
 - - - - - - - - - - - - - - - - - - - - - - - - O N E W A Y - - - - - - - - - - - - - - - - -

VARIABLE  SORTIES


MULTIPLE RANGE TEST


DUNCAN PROCEDURE
RANGES FOR THE .050 LEVEL -

           3.46  3.59

  THE RANGES ABOVE ARE TABULAR VALUES.
THE VALUE ACTUALLY COMPARED WITH  MEAN(J)-MEAN(I) IS..
      20.3825 * RANGE * SQRT(1/N(I) + 1/N(J))

  HOMOGENEOUS SUBSETS  (SUBSETS OF GROUPS, WHOSE HIGHEST AND LOWEST MEANS DO
NOT DIFFER BY MORE THAN THE SHORTEST SIGNFICANT RANGE FOR A
SUBSET OF THAT SIZE)

  SUBSET  1

  GROUP       GRP  1
MEAN      388.3333
- - - - - - - - - -

  SUBSET  2

  GROUP       GRP  2
MEAN      581.3333
- - - - - - - - - -

  SUBSET  3

  GROUP       GRP  3
MEAN      687.0000
- - - - - - - - - -
```

Fig. 5.2.2 Duncan's Test on MTBFs



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
shifted to left	478.0	505.0	553.0	512.0	38.0
symmetrical $\alpha = \beta = 2.0$	483.0	488.0	515.0	495.3	17.2
shifted to right (norm)	539.0	602.0	603.0	581.3	36.7

Fig. 5.3 Shape Parameter Sensitivity

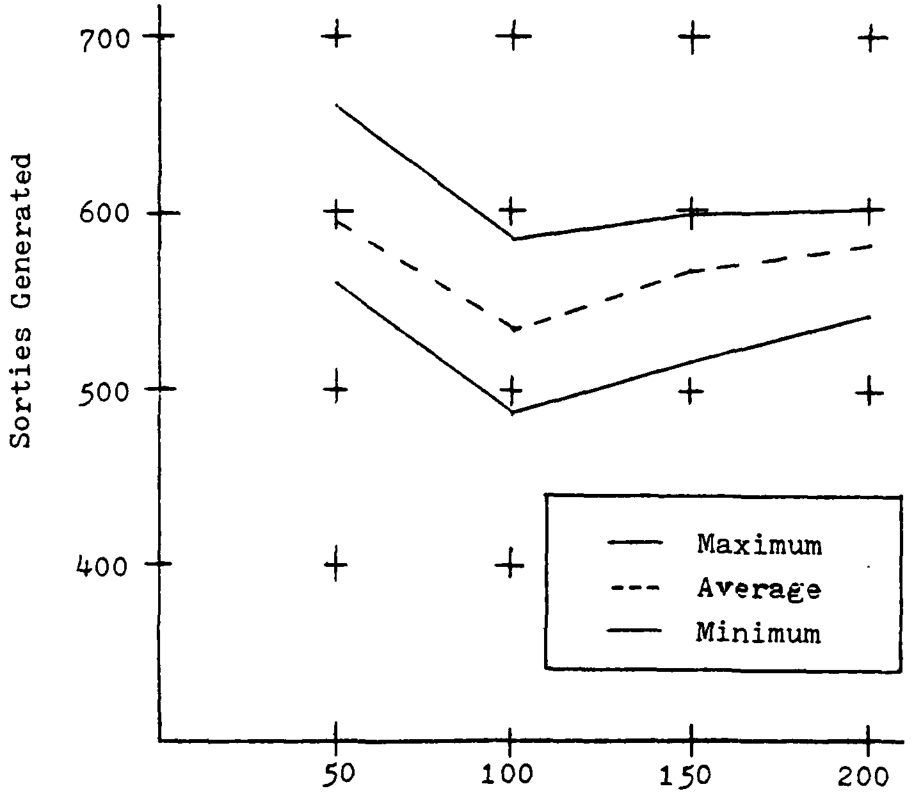
judgment of the reliability of systems by referring to the discussion of shape parameters in Appendix C.

Initial Value of Engine Run Time

Each aircraft is initially assigned a value for engine run time. The value is from a uniform distribution with user selectable limits. The uniform distribution was used to reflect a stable flow of aircraft through major overhaul operations at selected hours of engine operation. The levels selected were 50, 100, 150, and 200 hours. The normal value used in the model was 200 hours. Levels above 200 hours were not used because 200 is both a normal level and an outside limit. Results are depicted in Figure 5.4. No statistical difference existed between the four levels, although the 50-hour run had the highest mean, which was intuitively expected. The model was not very sensitive to this factor.

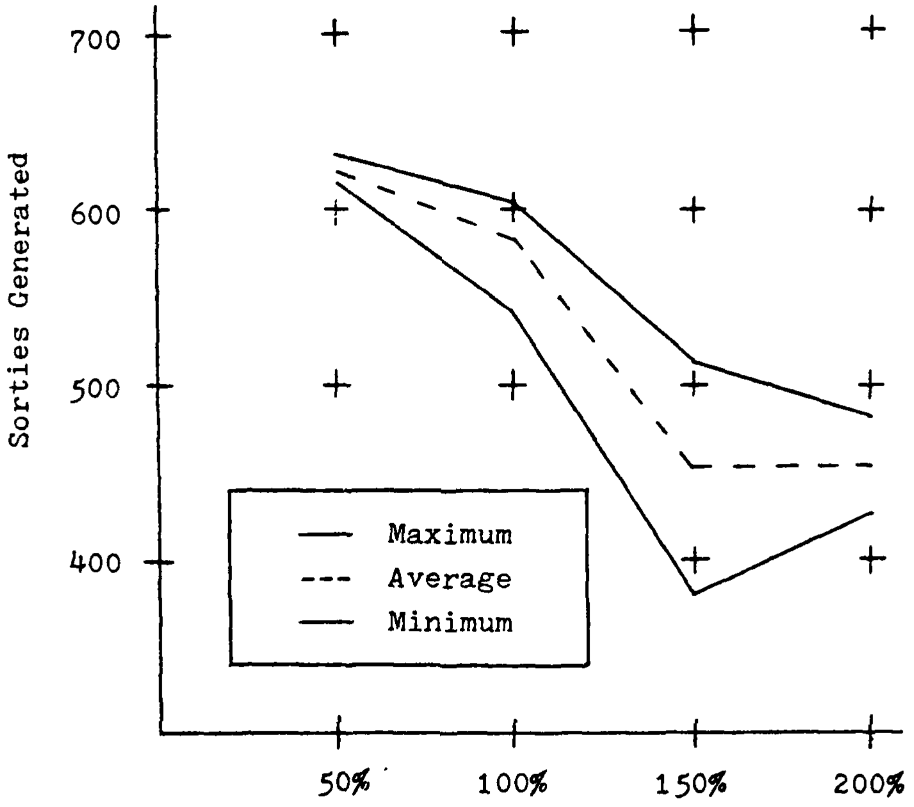
Attrition

Attrition rates were tested at half the normal values, at the normal values, at 1.5 times the normal values, and at 2.0 times the normal values. Results are depicted in Figure 5.5. The results were statistically different, with the means of the normal value run and the half normal value falling into another group. There appears to be a steep fall off in the response variable between the two groups. This may have been due to the effects of how the replacement squadrons were factored into the simulation at the higher attrition levels.



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
50 hours	560.0	572.0	657.0	596.3	52.9
100 hours	483.0	584.0	532.0	533.0	50.5
150 hours	599.0	514.0	584.0	565.7	45.4
200 hours (n)	539.0	602.0	603.0	581.3	36.7

Fig. 5.4 Initial Engine Run Time Sensitivity



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
50% of norm	628.0	615.0	621.0	621.3	6.5
100% of norm	539.0	602.0	603.0	581.3	36.7
150% of norm	467.0	377.0	512.0	452.0	68.7
200% of norm	426.0	457.0	481.0	454.7	27.6

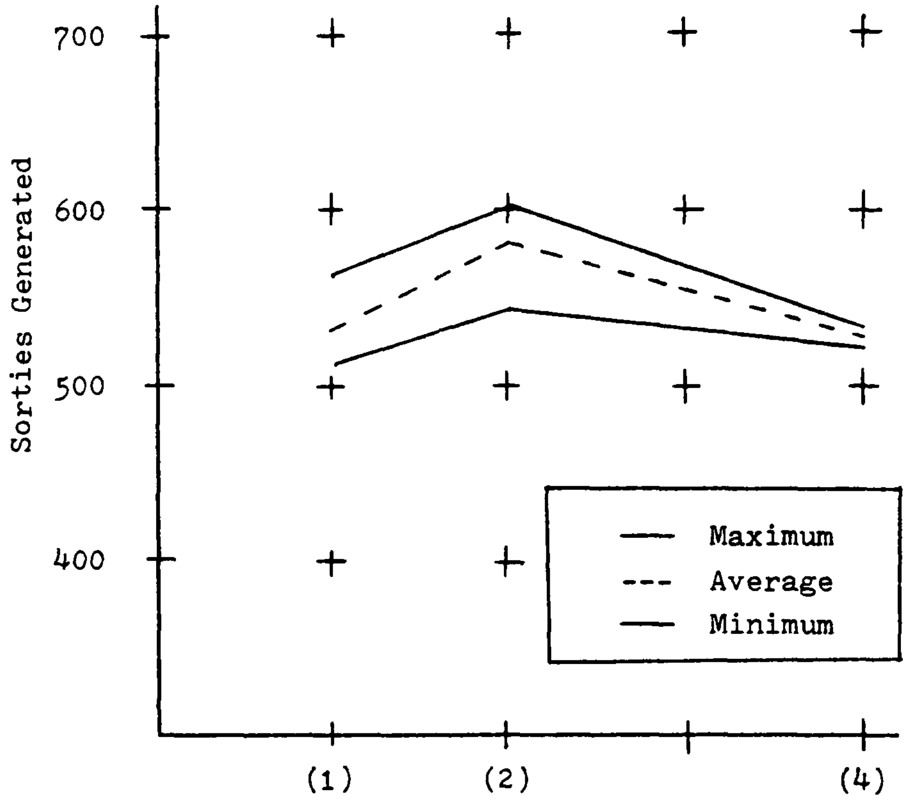
Fig. 5.5 Attrition Rate Sensitivity

Users should do more extensive sensitivity analysis on this variable, since the model appears to be sensitive to attrition rates within the ranges examined.

Geographical Mix of Fragmentary Order

Two different geographical mixes were tested against the normal frag. The normal frag configured more aircraft for Area 3 and two 24-ship gaggles were launched on Day 1, with a 20-minute delay in further launches after each gaggle departed (Appendix B, page 131). The rest of the aircraft in the normal frag were sent to Area 2 and Area 1. In one excursion, all aircraft were configured for Area 3, and a 36-ship and a 24-ship gaggle launched on Day 1 and Day 2, with a 20-minute delay in further launches after each gaggle departed. This clearly favored Area 3 missions, which are of longer duration with higher attrition rates. The other excursion configured all aircraft for Area 1 and sent all aircraft only to Area 1. Area 1 missions are the shortest duration with the lowest attrition rates. The results are shown in Figure 5.6.

The statistical test found that the two excursions produced similar values in the response variable. The values were lower than the normal. The test also groups the Area 3 excursion with the normal, which was probably due to both having much larger variances than the excursion which favored Area 1. The Area 1 excursion standard deviation was the lowest amount of variance observed in any experimentation with



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
(1) mostly to area three	560.0	511.0	521.0	530.7	25.9
(2) norm	539.0	602.0	603.0	581.3	36.7
(4) all a/c to area one	525.0	520.0	531.0	525.3	5.5

Fig. 5.6 Frag Geographical Mix Sensitivity

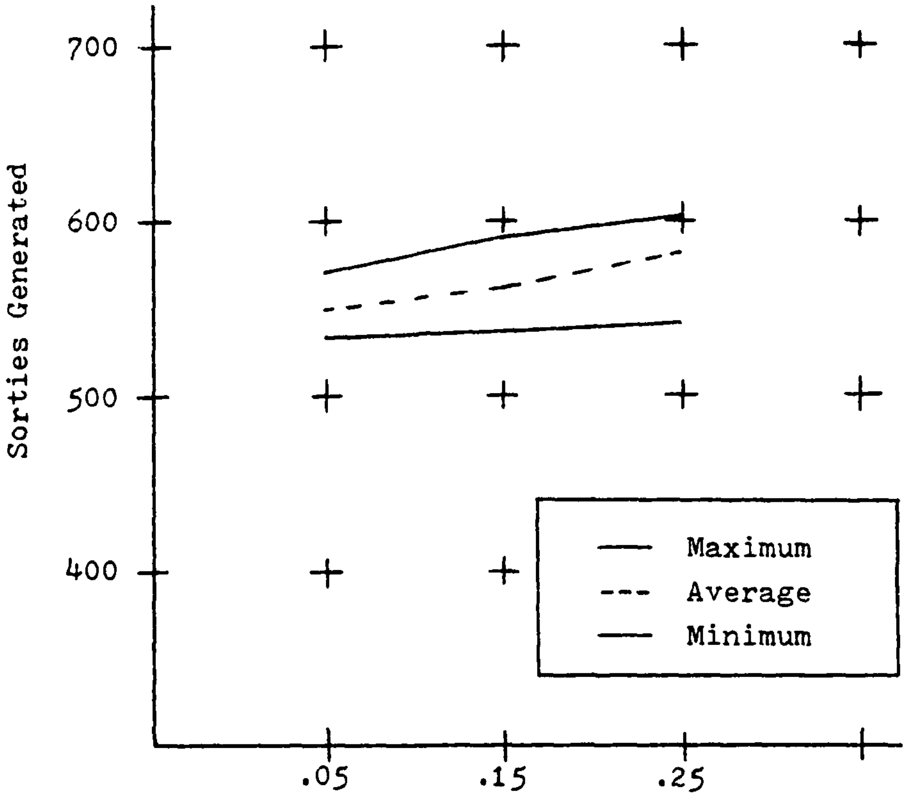
the model. No explanation for the results is obvious. The mean of the 28 normal runs was 549.7, with a standard deviation of 37.8. This is more in the realm of the excursion results. Possibly the three run normal test mean of 581.3 was a little high. In any case, the model does not appear to be very sensitive to changes in the geographic mix of missions in the frag.

Preflight Delay

Because the normal probability of preflight delay was relatively high, and because having a delay meant a little higher chance of a maintenance failure, this factor was also checked for sensitivity. (Incurring a delay at preflight causes the aircraft to fail if any system's NTOF is within a user specified number of minutes until failure.) Since the normal value was in the high range, only lower values were tested. The results are shown in Figure 5.7. There was no statistical difference in the means of the response variables at a 95 percent confidence level. The model is not very sensitive to this factor within the range of values tested.

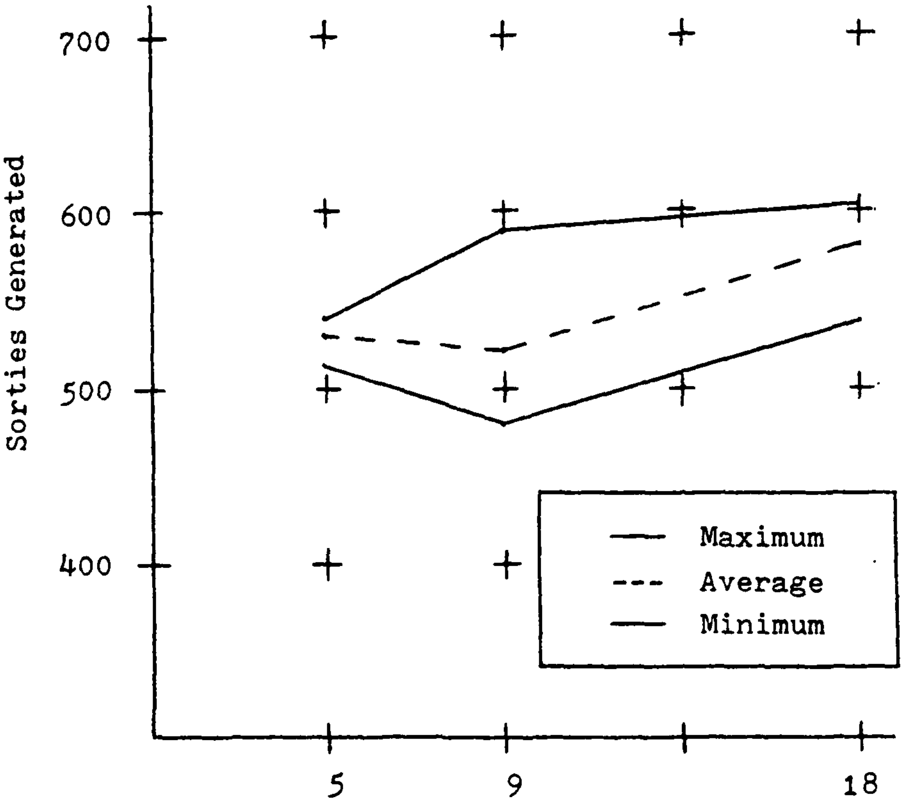
Rearming Crews

The normal number of arming crews was 18, which is on the high side. Only lower values, five and nine, were checked in the excursions. The means were not statistically different. The results are shown in Figure 5.8. The nearly level slope of the maximum curve between 9 and 18 crews agrees with what was expected. The normal maximum utilization of rearming



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
0.05	534.0	537.0	570.0	547.0	20.0
0.15	590.0	536.0	557.0	561.0	27.2
0.25 (norm)	539.0	602.0	603.0	581.3	36.7

Fig. 5.7 Preflight Delay Sensitivity



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
5 crews	534.0	540.0	514.0	529.3	13.6
9 crews	491.0	478.0	592.0	520.3	62.4
18 crews	539.0	602.0	603.0	581.3	36.7

Fig. 5.8 Rearming Crew Sensitivity

crews is around 10. This can be seen in Figure 4.4. Having more crews than the maximum number utilized has little consequence, as expected. In any event, the model is not very sensitive to the number of arming crews.

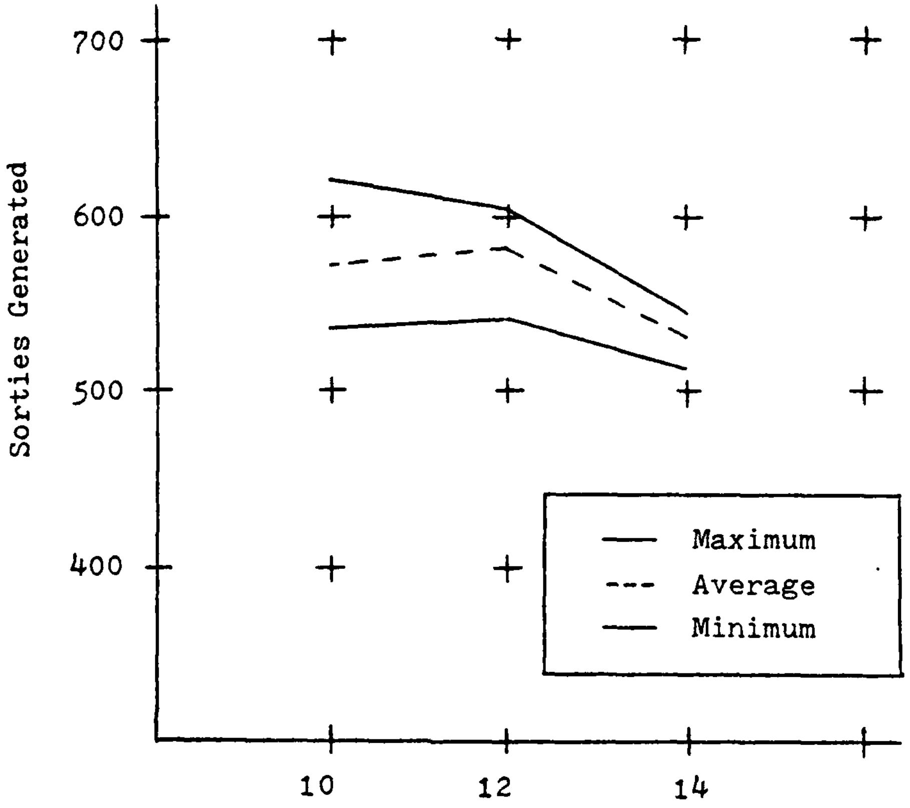
Replacement Squadron Trigger Level

Scheduling of a replacement squadron is triggered based on a user specified level of operational aircraft remaining in a squadron in the evening. The normal trigger point (variable LIMITAC) is 12 aircraft. The excursion values tested were 10 and 14. The results are shown in Figure 5.9. Statistically, there was no significant difference in the results at a 95 percent confidence level. This is only true within the tested range. Excursions outside this range may be significant. Users should perform further sensitivity analysis if the value of LIMITAC is set outside this range.

Screening Factors With a Fractional Factorial Experiment

Introduction

In order to determine which of the seven internal factors were the most critical, an experiment was required. A full factorial design was impractical for seven factors at only two levels of each of the factors (a 2^7 design). The 2^7 design required 128 runs per replication, which was infeasible given the computer resources available. An alternate method for initial screening of factors is use of a fractional



Test Levels	Data Pt. 1	Data Pt. 2	Data Pt. 3	Mean x	St.Dev. s
10 aircraft	549.0	620.0	535.0	568.0	45.6
12 (norm)	539.0	602.0	603.0	581.3	36.7
14 aircraft	513.0	541.0	541.0	531.7	16.2

Fig. 5.9 Replacement Squadron Trigger Level Sensitivity

factorial design.

The Fractional Factorial Design

To conduct a useful fractional factorial screening experiment, the main effects must be kept clear of aliasing. In addition, to have greater clarity in the results, it is useful to avoid aliasing main effects with second order effects. Aliasing between second order and higher effects was allowed because of the difficulty of physical explanation and the belief that these interactions could be assumed to be much smaller than the main and second order effects (Montgomery, 1976:239-251).

Another important consideration in experimental design is the number of levels per factor which will be examined. The minimum number of levels for experimentation is two. If the levels are properly chosen to reflect the upper and lower limits of levels possible for a factor, a screening experiment can provide useful results at only two levels per factor. For this reason, and to reduce the amount of computer resources required, only two levels were used in the screening experiment.

Using the above constraints results in a 2**7-3 fractional factorial design. This design keeps main and second order effects clear of aliasing, but does not prevent aliasing between second order effects. The specific design used was taken from Montgomery (1976). By using the defining equation:

$$I = ABCE = ABDF = ACDG = CDEF = BDEG = BDFG = AEFG$$

the design in Figure 5.10 was developed using the method defined

Factors		Low (-)	High (+)
A	Mx Teams	36	78
B	Wing Shops	0	4
C	Shelters	18	42
D	Trucks	12	40
E	Hotpits	0	4
F	MMT Units	1	4
G	Sqdn Shops	1	8

Run		A	B	C	D	E	F	G
1	(1)	-	-	-	-	-	-	-
2	aefg	+	-	-	-	+	+	+
3	bef	-	+	-	-	+	+	-
4	abg	+	+	-	-	-	-	+
5	ceg	-	-	+	-	+	-	+
6	acf	+	-	+	-	-	+	-
7	bcfg	-	+	+	-	-	+	+
8	abce	+	+	+	-	+	-	-
9	dfg	-	-	-	+	-	+	+
10	ade	+	-	-	+	+	-	-
11	bdeg	-	+	-	+	+	-	+
12	abdf	+	+	-	+	-	+	-
13	cdef	-	-	+	+	+	+	-
14	acdg	+	-	+	+	-	-	+
15	bcd	-	+	+	+	-	-	-
16	abcdefg	+	+	+	+	+	+	+

Fig. 5.10 The Fractional Factorial Design

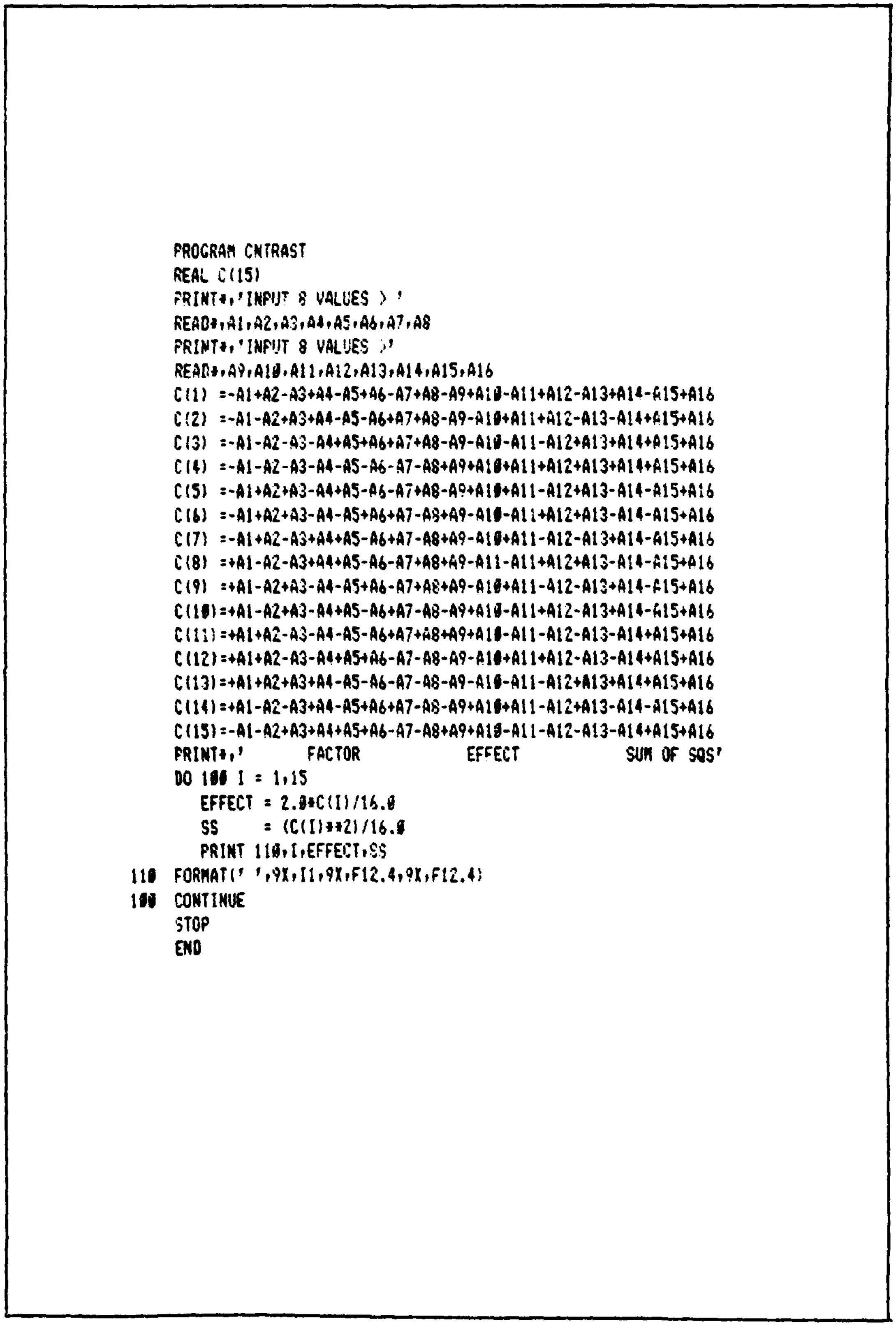
in the Montgomery text. This design determines the 16 runs necessary to execute the design. The minus sign (-) means use the lower level of a factor, while the plus sign (+) means use the higher level. This design will allow the significant effects to be established with a minimum number of runs.

Selecting Levels of Factors

The levels of each factor were chosen to reflect the possible upper and lower limits of the respective factors. The lower level was chosen as the reasonable minimum value for the factor. In the case of Wing Maintenance shops, for example, this was zero. To allow some maintenance to continue on major problems, and because zero was not a reasonable minimum for MMT units, the lowest level of MMT units was chosen as one. The higher levels (+) were established to be the highest reasonable levels which could ever be expected. The levels chosen are depicted in Figure 5.10.

Analysis of Results

The 16 data points were analyzed using the method of contrasts (Montgomery, 1976:193-197). This procedure involves plotting the estimates of effects for each factor on normal probability paper. To aid in analyzing the plot, all the estimates of effects were plotted, not just the estimates of effects for the main factors. The estimates of effects were calculated using the program depicted in Figure 5.11. The estimates are calculated by adding or subtracting the response variables to arrive at a total of response variables for a



```
      PROGRAM CNTRAST
      REAL C(15)
      PRINT*,'INPUT 8 VALUES > '
      READ*,A1,A2,A3,A4,A5,A6,A7,A8
      PRINT*,'INPUT 8 VALUES >'
      READ*,A9,A1Ø,A11,A12,A13,A14,A15,A16
      C(1) =-A1+A2-A3+A4-A5+A6-A7+A8-A9+A1Ø-A11+A12-A13+A14-A15+A16
      C(2) =-A1-A2+A3+A4-A5-A6+A7+A8-A9-A1Ø+A11+A12-A13-A14+A15+A16
      C(3) =-A1-A2-A3-A4+A5+A6+A7+A8-A9-A1Ø-A11-A12+A13+A14+A15+A16
      C(4) =-A1-A2-A3-A4-A5-A6-A7-A8+A9+A1Ø+A11+A12+A13+A14+A15+A16
      C(5) =-A1+A2+A3-A4+A5-A6-A7+A8-A9+A1Ø+A11-A12+A13-A14-A15+A16
      C(6) =-A1+A2+A3-A4-A5+A6+A7-A8+A9-A1Ø-A11+A12+A13-A14-A15+A16
      C(7) =-A1+A2-A3+A4+A5-A6+A7-A8+A9-A1Ø+A11-A12-A13+A14-A15+A16
      C(8) =+A1-A2-A3+A4+A5-A6-A7+A8+A9-A11-A11+A12+A13-A14-A15+A16
      C(9) =+A1-A2+A3-A4-A5+A6-A7+A8+A9-A1Ø+A11-A12-A13+A14-A15+A16
      C(1Ø)=+A1-A2+A3-A4+A5-A6+A7-A8-A9+A1Ø-A11+A12-A13+A14-A15+A16
      C(11)=+A1+A2-A3-A4-A5-A6+A7+A8+A9+A1Ø-A11-A12-A13-A14+A15+A16
      C(12)=+A1+A2-A3-A4+A5+A6-A7-A8-A9-A1Ø+A11+A12-A13-A14+A15+A16
      C(13)=+A1+A2+A3+A4-A5-A6-A7-A8-A9-A1Ø-A11-A12+A13+A14+A15+A16
      C(14)=+A1-A2-A3+A4-A5+A6+A7-A8-A9+A1Ø+A11-A12+A13-A14-A15+A16
      C(15)=-A1-A2+A3+A4+A5+A6-A7-A8+A9+A1Ø-A11-A12-A13-A14+A15+A16
      PRINT*,'       FACTOR            EFFECT            SUM OF SQS'
      DO 1ØØ I = 1,15
         EFFECT = 2.Ø*C(I)/16.Ø
         SS     = (C(I)**2)/16.Ø
         PRINT 11Ø,I,EFFECT,SS
11Ø   FORMAT(' ',9X,I1,9X,F12.4,9X,F12.4)
1ØØ   CONTINUE
      STOP
      END
```

Fig. 5.11 Estimates of Effects Calculations

given column in Figure 5.10. The response variable is added if there is a (+) in the column, and it's subtracted if there is a (-). The resulting total is called the contrast for that particular effect. The contrast is multiplied by two (levels) and divided by 16 (number of data points) to produce the estimate of effect for each factor. The results are shown in Figure 5.12.

The next step in the method of contrasts is to order the estimates of effects from the lowest to the highest value. The ordering is shown in Figure 5.12 as ascending order. After ordering the estimates, the results are plotted on normal probability paper. On the probability scale of the normal probability paper, the value of (J-.5)/16 is plotted, where J is the order number given in the ascending order column of Figure 5.12. The value of the estimate of effect is plotted using the linear scale.

After the estimates of effects are plotted, those points that lie significantly off the line formed by the plot are significant. Insignificant effects tend to fall along a straight line on the normal probability paper. The plots are shown in Figure 5.13. A line has been added to aid in identifying the significant points (Montgomery, 1976:195). The points representing effects A and F are significant, as well as the sum of interactions of the aliased second order effects AB = CE = DF. Effect A corresponds to MXTEAMS, and Effect F corresponds to MMT units. Effect AB = CE = DF corresponds to the sum of the interactions of MXTEAMS with wing shops, shelters with Hot

$$EST = (2*CONTRAST)/16$$

$$SS = (CONTRAST)^2/16$$

Effect	Estimation of Effect	Sum of Squares	Ascending Order
A	132.88	70623	13
B	-4.88	95	5
C	-4.13	68	6
D	-86.13	29670	2
E	1.63	10	8
F	240.13	230640	15
G	63.38	16065	12
AB=CE=DF	142.63	81367	14
AC=BE=DG	-33.63	4522	3
AD=BF=CG	3.88	60	9
AE=BC=FG	-1.38	7	7
AF=BD=EG	-104.38	43576	1
AG=CD=EF	-22.63	2047	4
BG=CF=DE	31.63	4000	11
ABG=...	13.13	689	10

Fig. 5.12 Estimates of Effects Results

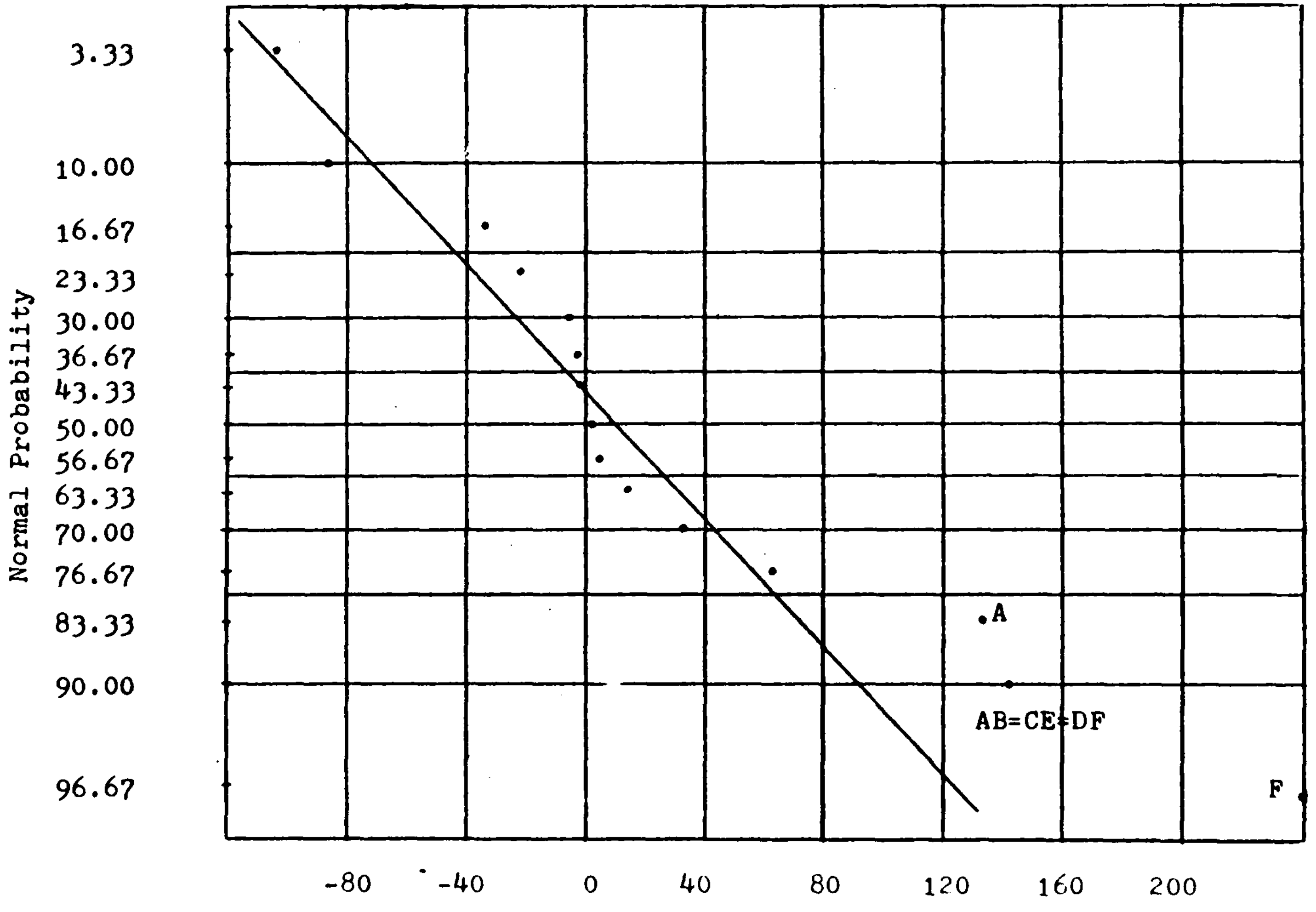


Fig. 5.13 Method of Contrasts Normal Probability Plot

Pit hydrants, and fuel trucks with MMT units.

The results indicate a need for closer investigation of the MXTEAMS and MMT unit factors. The significant interaction effect is more difficult to deal with. The DF interaction has no physical meaning, so it can be assumed to be negligible in contributing to the effect. The CE interaction has a physical meaning. It may have contributed to the significance of the effect when shelter refueling was closed down, Hot Pits were at zero, and fuel trucks were at 12. The response variable for that particular series of runs had a mean of 536 with a standard deviation of 58.9. This is a normal range response. In addition, reducing fuel trucks to a level where their influence is greatly felt, even with shelter and Hot Pit refueling shut down, would be a difficult task on a real airfield. Therefore, these factors are not of great interest. MXTEAMS and wing shops do not interact directly; however, since wing shops do work on the same type of problems repaired by MMT units, a closer look at wing shops would be valuable. With the above results in mind, a full factorial experiment was designed to test three factors--MMT units, MXTEAMS, and Wing Maintenance shops.

The Full Factorial Experiment

Factor Levels and Replications

The factors which were not significant in the screening experiment were reset to their normal scenario values for this experiment. The same low (-) and high (+) levels as in the screening experiment were used for MMT units, MXTEAMS, and wing shops. Only two levels were used, once again, to conserve computer resources. A 3**3 design (three levels) requires 27 runs per replication. The 2**3 (two levels) design used required only 8 runs per replication.

With the design established, the number of replications had to be fixed. The large variance of the model dictated that as many runs be made per cell as possible. A tradeoff was made between cost and variance reduction, and seven replications (seven runs per cell) were selected as shown in Figure 5.14.

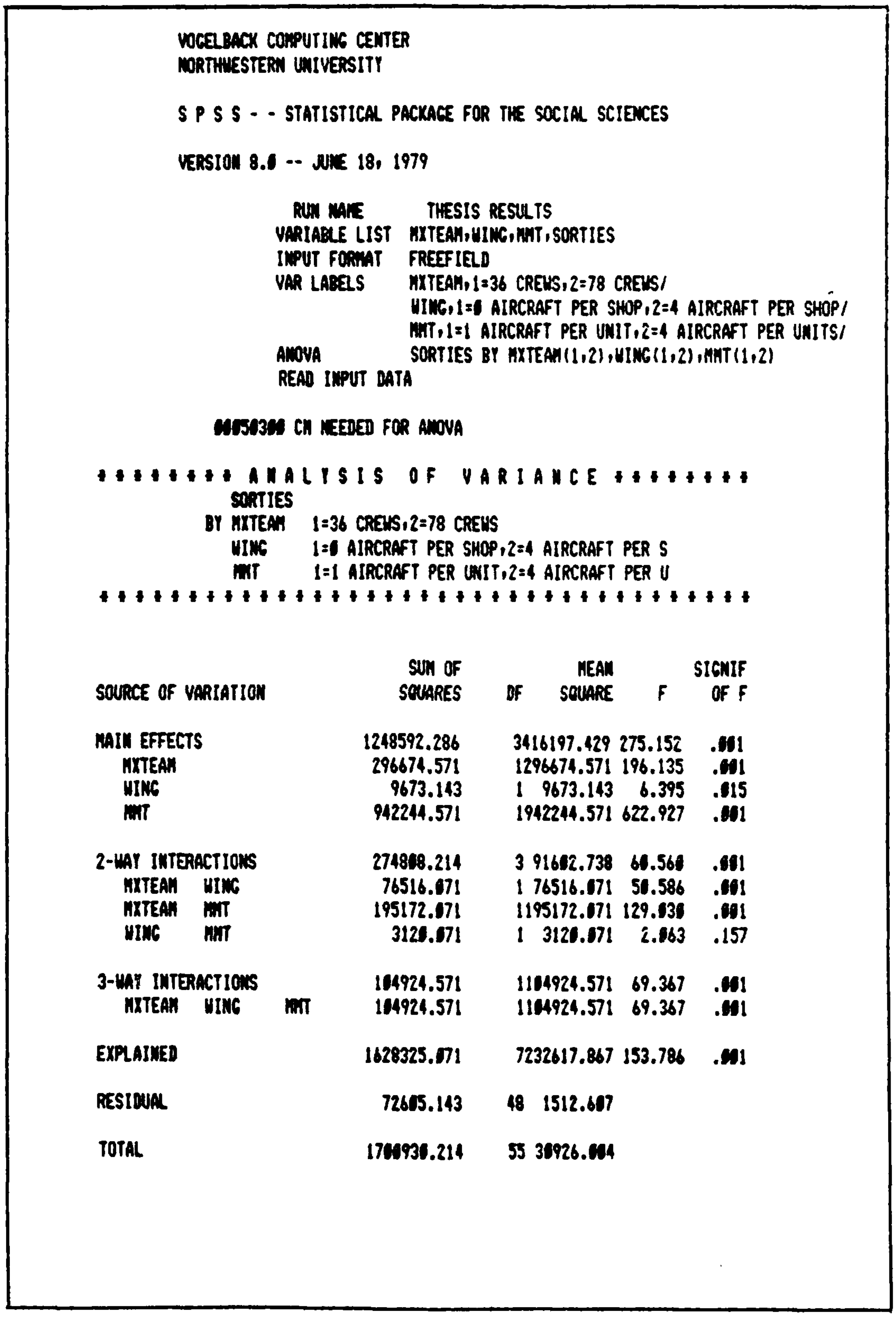
Analysis of Full Factorial Results

After the seven replications had been completed, SPSS was used to do an ANOVA. The results are shown in Figure 5.15. The results show that the main effects of MMT units and MXTEAMS were significant at a 99 percent confidence level. The effects of wing shops were significant at a lower level (85 percent). The second order interactions were significant at similar confidence levels. The three-way interactions also showed as highly significant.

In an attempt to further define which of these main effects was most significant, SPSS was used to run Duncan's Multiple Range Tests at the 95 percent confidence level on each significant effect. The results are shown in Figure 5.16. The tests verified that MMT units and MXTEAMS main effects were indeed significant, while wing shops were

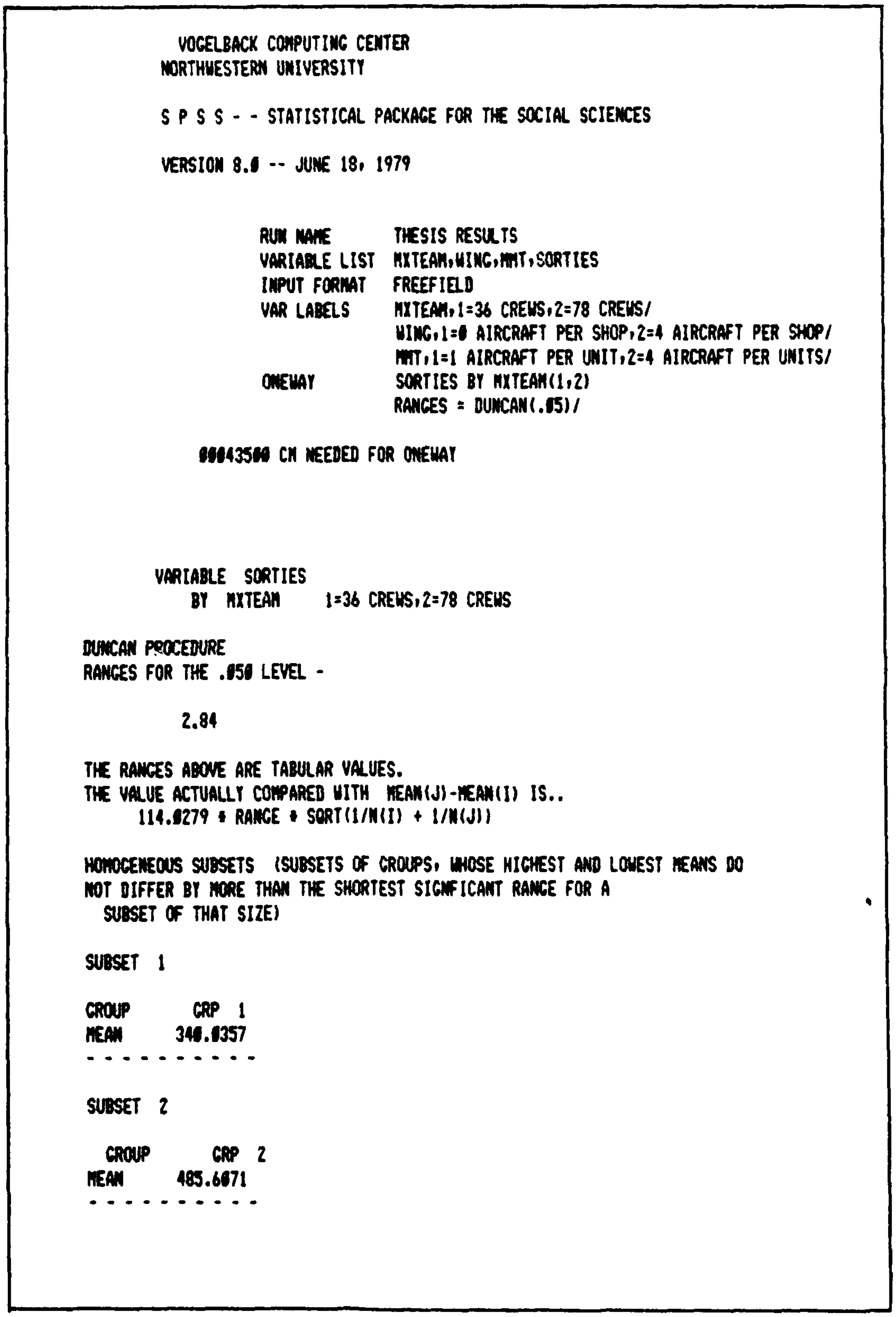
2^3 Full Factorial	1 MMT Unit		4 MMT Units	
	0 Wing Shops	4 Wing Shops	0 Wing Shops	4 Wing Shops
36 MxTeams	XXXXXXX	XXXXXXX	XXXXXXX	XXXXXXX
78 MxTeams	XXXXXXX	XXXXXXX	XXXXXXX	XXXXXXX

Fig. 5.14 The Full Factorial Design



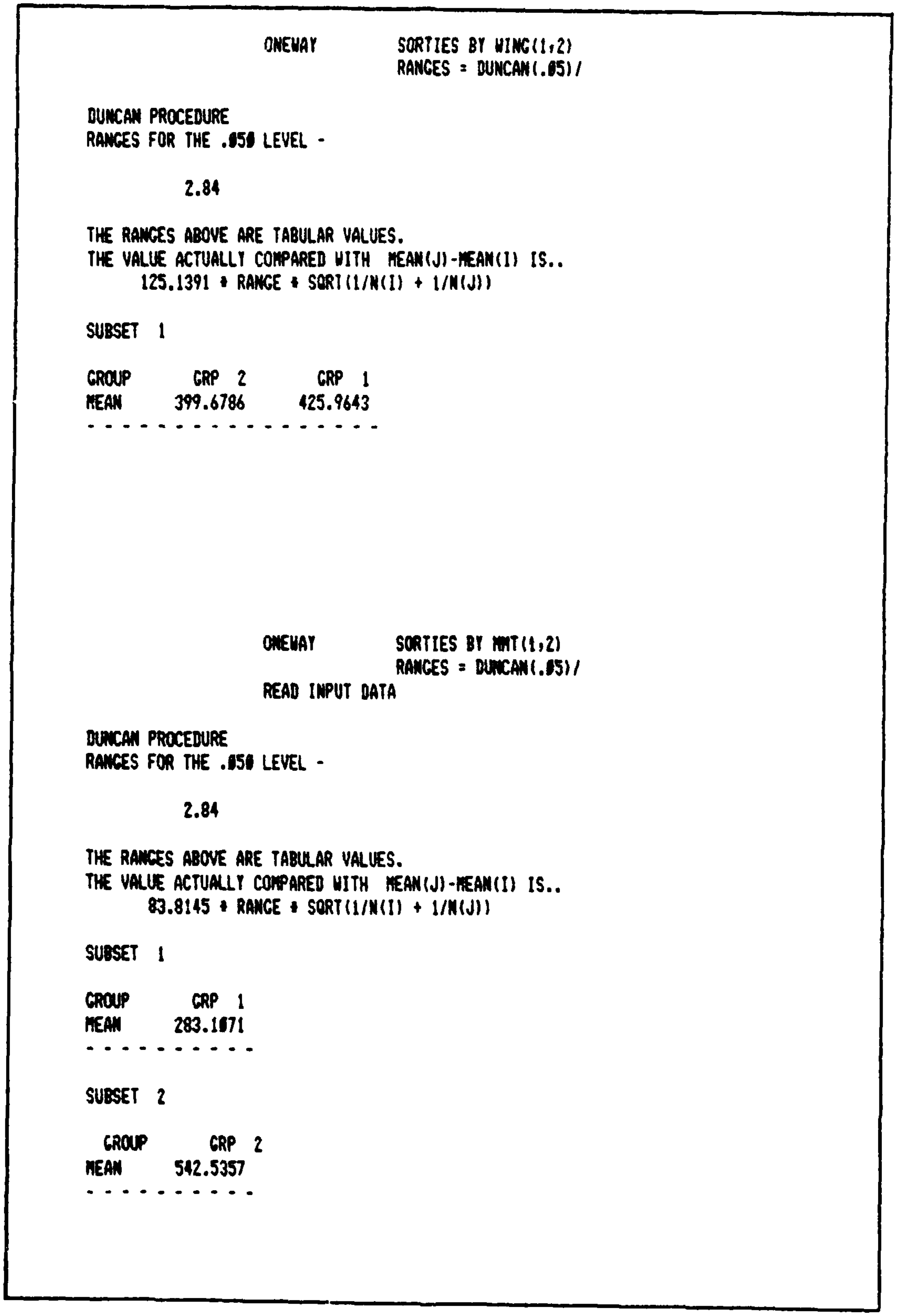
```
            VOGELBACK COMPUTING CENTER
            NORTHWESTERN UNIVERSITY

            S P S S - - STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES

            VERSION 8.0 -- JUNE 18, 1979

                    RUN NAME       THESIS RESULTS
                   VARIABLE LIST  MXTEAM,WING,MMT,SORTIES
                   INPUT FORMAT   FREEFIELD
                   VAR LABELS     MXTEAM,1=36 CREWS,2=78 CREWS/
                                  WING,1=0 AIRCRAFT PER SHOP,2=4 AIRCRAFT PER SHOP/
                                  MMT,1=1 AIRCRAFT PER UNIT,2=4 AIRCRAFT PER UNITS/
                   ANOVA          SORTIES BY MXTEAM(1,2),WING(1,2),MMT(1,2)
                   READ INPUT DATA

              00050300 CM NEEDED FOR ANOVA

      * * * * * * * * A N A L Y S I S   O F   V A R I A N C E * * * * * * * *
                 SORTIES
              BY MXTEAM   1=36 CREWS,2=78 CREWS
                 WING     1=0 AIRCRAFT PER SHOP,2=4 AIRCRAFT PER S
                 MMT      1=1 AIRCRAFT PER UNIT,2=4 AIRCRAFT PER U
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


                                      SUM OF              MEAN          SIGNIF
      SOURCE OF VARIATION            SQUARES      DF     SQUARE     F      OF F

      MAIN EFFECTS                 1248592.286     3416197.429 275.152   .001
         MXTEAM                     296674.571     1296674.571 196.135   .001
         WING                         9673.143     1  9673.143   6.395   .015
         MMT                        942244.571     1942244.571 622.927   .001

      2-WAY INTERACTIONS            274808.214     3 91602.738  60.560   .001
         MXTEAM   WING               76516.071     1 76516.071  50.586   .001
         MXTEAM   MMT               195172.071     1195172.071 129.030   .001
         WING     MMT                 3120.071     1  3120.071   2.063   .157

      3-WAY INTERACTIONS            104924.571     1104924.571  69.367   .001
         MXTEAM   WING     MMT      104924.571     1104924.571  69.367   .001

      EXPLAINED                    1628325.071     7232617.867 153.786   .001

      RESIDUAL                       72605.143    48  1512.607

      TOTAL                        1700930.214    55 30926.004
```

Fig. 5.15 Full ANOVA Results



```
          VOGELBACK COMPUTING CENTER
         NORTHWESTERN UNIVERSITY

         S P S S - - STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES

         VERSION 8.0 -- JUNE 18, 1979


                  RUN NAME       THESIS RESULTS
                  VARIABLE LIST  MXTEAM,WING,MMT,SORTIES
                  INPUT FORMAT   FREEFIELD
                  VAR LABELS     MXTEAM,1=36 CREWS,2=78 CREWS/
                                 WING,1=0 AIRCRAFT PER SHOP,2=4 AIRCRAFT PER SHOP/
                                 MMT,1=1 AIRCRAFT PER UNIT,2=4 AIRCRAFT PER UNITS/
                  ONEWAY         SORTIES BY MXTEAM(1,2)
                                 RANGES = DUNCAN(.05)/

            00043500 CM NEEDED FOR ONEWAY




        VARIABLE  SORTIES
           BY  MXTEAM     1=36 CREWS,2=78 CREWS

DUNCAN PROCEDURE
RANGES FOR THE .050 LEVEL -

          2.84

THE RANGES ABOVE ARE TABULAR VALUES.
THE VALUE ACTUALLY COMPARED WITH  MEAN(J)-MEAN(I) IS..
     114.0279 * RANGE * SQRT(1/N(I) + 1/N(J))

HOMOGENEOUS SUBSETS  (SUBSETS OF GROUPS, WHOSE HIGHEST AND LOWEST MEANS DO
NOT DIFFER BY MORE THAN THE SHORTEST SIGNFICANT RANGE FOR A
  SUBSET OF THAT SIZE)

SUBSET  1

GROUP       GRP  1
MEAN     340.0357
- - - - - - - - - -

SUBSET  2

  GROUP       GRP  2
MEAN     485.6071
- - - - - - - - - -
```

Fig. 5.16.1 Duncan's Test on Significant Effects



```
                  ONEWAY        SORTIES BY WING(1,2)
                                RANGES = DUNCAN(.05)/

DUNCAN PROCEDURE
RANGES FOR THE .050 LEVEL -

          2.84

THE RANGES ABOVE ARE TABULAR VALUES.
THE VALUE ACTUALLY COMPARED WITH  MEAN(J)-MEAN(I) IS..
     125.1391 * RANGE * SQRT(1/N(I) + 1/N(J))

SUBSET  1

GROUP       GRP  2       GRP  1
MEAN      399.6786     425.9643
- - - - - - - - - - - - - - - - -







                  ONEWAY        SORTIES BY MNT(1,2)
                                RANGES = DUNCAN(.05)/
                  READ INPUT DATA

DUNCAN PROCEDURE
RANGES FOR THE .050 LEVEL -

          2.84

THE RANGES ABOVE ARE TABULAR VALUES.
THE VALUE ACTUALLY COMPARED WITH  MEAN(J)-MEAN(I) IS..
      83.8145 * RANGE * SQRT(1/N(I) + 1/N(J))

SUBSET  1

GROUP       GRP  1
MEAN      283.1071
- - - - - - - - - -

SUBSET  2

  GROUP       GRP  2
MEAN      542.5357
- - - - - - - - - -
```

Fig. 5.16.2 Duncan's Test on Significant Effects

significant to a lesser degree. The results of this experiment support the indications of the fractional factorial screening experiment. However, it was not possible to decide which factor was most significant between MMT units and MXTEAMS.

VI. CONCLUSIONS AND RECOMMENDATIONS

Introduction

The remainder of this chapter is divided into two sections. The first section contains conclusions reached through this study effort. The other section contain recommendations for further study of the airfield attack problem using the airfield model.

The approach presented in this thesis for identifying critical airfield target elements has proven to have much greater conceptual clarity than previous approaches. Traditionally, an airfield model was built and then the airfield was attacked using a Monte Carlo simulation. Damage was assessed and then the airfield model was changed to reflect the results of the attack. The changes to the model were in the form of number of servers or channels, parameters of probability distributions, or the distributions themselves. The model was then executed. This is the way the TSARINA/TSAR combination operates. It is the conceptual way that AIRBASE functions in that the user must specify when an attack will occur and the results. These inputs must be generated by the user. When AIRBASE is used at Eglin, there is a damage assessment program available for that purpose.

By eliminating the conceptual attack, and by only trying to determine which factors are key elements, there is cnly one level of abstraction as opposed to the two levels inprevious approaches. This makes the model much easier to experiment with, and it seems to be an attractive approach to the airfield targeting problem.

In reviewing the objectives listed in Chapter I, the authors believe each of the points has been covered adequately for the purposes of this thesis. The model is available and well documented to provide for ease of use. The methodology to determine criticality is explicitly developed and demonstrated. The model and methodology can now be taken the final mile. All the input values currently in the model are the best guesses of one of the authors. Each of the values and probabilities were selected to be in the reasonable range. They were reviewed by another experienced fighter pilot, and amended slightly prior to use in the normal scenario. Prior to experimentation by another user, all the values will have to be updated with better data. Other than changing the values, the methodology remains the same.

With respect to the Problem Statement in Chapter I, this thesis presents a methodology which can determine the critical elements on an airfield. This thesis did not determine those elements due to the lack of accurate data for input. Users with accurate data should be able to determine a first-cut criticality for those elements included in the present model. This will be addressed in the following section. Refer to Annex A for additional remarks.

Conclusions

The results of the experiments presented in Chapter V were entirely consistent with what could be expected given the input values for the scenario. The fact that the maintenance sections which repaired serious (level four and five) failures were among the most critical factors was not surprising. Having MXTEAMS (crew chiefs) show up as a critical factor was not surprising given their importance in the launch and recovery/turnaround processes. While the results were not surprising, that is not to say they were expected or predicted. The model is complex enough that it was not possible to predict which factors would be identified by the experimentation.

The experimental results were consistent with the inputs to the model. The experiments showed MMT units were more significant than wing shops. A review of the input service times in Appendix B, pages 138 and 139, probably shows why. The MMT units were given very much shorter service times than wing shops. This made them much more significant in returning aircraft to service even though they did not concurrently repair level two and three problems.

The methodology employed in this study was adequate to point out the relative significance of two out of three important factors. The methodology used was conceptually straightforward. The same approach could be used on any other complex system which can be modeled with a single response variable. The airfield model should provide a useful analytical tool for studying the airfield attack problem in the future.

Recommendations for Further Study

During the course of building the model, experimentation, and writing up the thesis several ideas for improvements to the model surfaced. The most significant of these ideas will be presented, but they are not presented in any order of significance. See Annex A for additional remarks.

The model could be improved with a structural change which would allow squadron maintenance and MMT units to work on aircraft without the aircraft having acquired an MXTEAM. Aircraft acquire an MXTEAM in the Maintenance Control subsection of the SLAM network (Appendix A, page 101) at node SPMX. If no MXTEAMs are available, the aircraft await one at SPMX before being processed on to SMXC, where they can then branch to MMT or squadron maintenance. One of the reasons that MXTEAMs showed up so significantly in the experiments is probably at least partly due to the fact that MMT and squadron maintenance could only work on aircraft which had a crew chief.

Another structural change should be made to add air-to-ground missile build-up, handling, loading, and maintenance. The same addition could be done for air-to-air missiles, or the two additions could be handled together.

A more sophisticated technique should be used to allocate the runway resource. Landing aircraft should continue to have priority for the runway, but the priority should be a function of their fuel state. In wartime, no commander wants to hold a flight on the ground to land one aircraft--unless the landing aircraft is minimum fuel. In addition, the landing aircraft are much safer while airborne in their natural element than on the ground. While contemplating a change to this area, the possible addition of formation landings should be considered for aircraft in the same flight. Conceptually, this is a relatively important area. Aircraft would never be allowed to pile up on the ground waiting for takeoff. When formation landings are convenient, they would certainly be used to expedite recovery and return to shelters/revetments.

A more detailed mission section could be written which would model different profiles to the different geographical areas. Fuel could be handled continuously. This would compliment the suggested routine for handling allocation of the runway using fuel as one of the decision variables. This mission section could be very extensive, but it only really requires what is necessary to place the aircraft in states where they can be processed as necessary by the model after a mission.

A central fuel storage area, or areas, could be added and a capability to resupply the airfield. These additions should only be made if these elements could be modeled with some degree of precision.

If any of the additions mentioned above are put into the model, the current AFIT SLAM limits will have to be raised above 500 nodes and 100 files. With that exception, there should be no problems with the additions.

In addition to improving the model, there is much that can be done in experimentation with the model. Time did not permit really extensive, in-depth sensitivity analysis. For example, in the area of attrition, it is almost certain that the way the replacement squadrons are scheduled into the airfield has an effect. There was insufficient time to find out how those factors interact. The area of MTBF sensitivity should also be more extensively examined. Also, each of the factors which wasn't very sensitive should be studied in detail to find out why the model wasn't very sensitive to that factor. This type of experimentation could be very valuable in understanding both the model and the airfield element interactions.

BIBLIOGRAPHY

AIRBASE Model Working Paper. Bethpage, NY: Grumman Aerospace Corporation, 6 May 1981.

Air Force Armament Laboratory AIRBASE Model. Chicago: A.T. Kearney, Inc., February 1978.

An Investigation of the Effects on Sortie Rate of Schedule, Maintenance and Basing Using the AIRBASE Simulation Model. Air Force Flight Dynamics Laboratory (AFFDL) Technical Report 79-3018. Chicago: A.T. Kearney, Inc., November 1978.

Emerson, Donald E. AIDA: An Airbase Damage Assessment Model. Rand Report R-1872-PR. Santa Monica, CA: The Rand Corporation, September 1976.

________. TSARINA -- User's Guide to a Computer Model for Damage Assessment of Complex Airbase Targets. Rand Report N-1460-AF. Santa Monica, CA: The Rand Corporation, July 1980.

Hoeber, Francis P. Military Applications of Modeling: Selected Case Studies. New York: Gordon and Breach Science Publishers, 1981.

Hokanson, Lawrence D. "Airfield Bomb Damage Repair," The Military Engineer, September/October 1975, pp. 260-262.

Law, Averill M. and W. David Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill Book Company, 1982.

Montgomery, Douglas S. Design and Analysis of Experiments. New York: John Wiley & Sons Publishing Co., 1976.

Nie, Normal H., et al. Statistical Package for the Social Sciences (SPSS). (Second Edition). New York: McGraw-Hill Book Company, 1982.

Pritsker, A. Alan B. Modeling and Analysis Using Q-GERT Networks (Second Edition). New York: John Wiley & Sons Publishing Co., 1979.

________ and Claude Dennis Pegden. Introduction to Simulation and SLAM. New York: John Wiley & Sons Publishing Co., 1979.

Shannon, Robert E. Systems Simulation: The Art and Science. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

Southard, Sara. Analyst, Joint Studies Group (TAC), Nellis AFB, Nevada 89191. Telephone interview, 19 May 1981; meeting, 17 July 1981.

Walpole, Ronald E. and Raymond H. Myers. Probability and Statistics for Engineers and Scientists (Second Edition). New York: Macmillan Publishing Co., Inc., 1978.

VITA

Robert W. Mann was born on 9 October 1944 in Taft, California. He graduated from high school in Alexandria, Virginia in 1962 and, after enlisting in the Air Force Reserve, attended the USAF Academy Preparatory School, USAFA, Colorado until May 1963. He received a Bachelor of Science degree and a commission from the Air Force Academy on 7 June 1967. After completion of pilot training in 1968, he served two tours in Southeast Asia, where he accumulated 1000 combat hours in the 0-2A (Covey) and the F-4E. After four years of flying F-4Ds in the U.K., he completed Imagery Intelligence School and the Targeting Officer's Course in 1976 at Lowry AFB, Colorado. He was the Wing Targeting Officer for the 474 TFW (F-111A, F-4D, F-16), Nellis AFB, Nevada, until entering the School of Engineering, Air Force Institute of Technology in August 1980.

Permanent Address: General Delivery
Las Vegas, Nevada

VITA

Brian J. Shook was born in Grand Rapids, Michigan on March 25, 1958. He graduated from high school in Caledonia, Michigan in 1976 and attended Michigan Technological University, from which he graduated in 1980 with a Bachelor's degree in Computer Science and a commission in the United States Air Force. He entered active duty and the Air Force Institute of Technology in August 1980.

Permanent Address: 300 Pleasant Ave.
Caledonia, MI 49316

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/GST/OS/82M-9

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A USER DEFINABLE SLAM AIRFIELD MODEL DESIGNED FOR EXPERIMENTATION AND ANALYSIS (Volume I)

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Robert W. Mann, Maj, USAF
Brian J. Shook, 2Lt, USAF

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Air Force Institute of Technology (AFIT/EN)
Wright-Patterson AFB, OH 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Air Force Institute of Technology (AFIT/EN)
Wright-Patterson AFB, OH 45433

12. REPORT DATE
March 1982

13. NUMBER OF PAGES
142

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17
4 JUN 1982
4 JUN 1982
Lynn Wolaver
LYNN E. WOLAVER
Dean for Research and Professional Development
AIR FORCE INSTITUTE OF TECHNOLOGY (ATC)
WRIGHT-PATTERSON AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Air Force Facilities
Airports
Computerized Simulation
Targets
Airfield
Airbase
Model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research effort was undertaken to investigate a methodology for determining the most critical elements on a fighter-bomber airbase with respect to sorties generated over a three-day period. The methodology is founded on a user definable computer simulation model written in SLAM (FORTRAN based) and supported by several FORTRAN routines. The remainder of the

Block 20:

methodology entails the use of factorial experimental designs for examining airfield element criticality. The airfield elements are the experimental factors. They are set to user specified levels according to the experimental design. The model produces a single response variable--sorties generated over a three-day period. Results are analyzed with common statistical techniques (Method of Contrasts, ANOVA, Duncan's Multiple Range Test). Special attention was placed on documentation of the model to insure ease of implementation by a user. Model usage is demonstrated with two experiments and their analysis. Because this methodology does not require Monte Carlo simulation of damage to the airfield, the determination of element criticality is straightforward. The lucrative targets on the airfield are then the most critical elements which can be effectively attacked with available weapons and delivery systems.